



1. Hoofdstuk 1

a) Welke belangrijke obstakels en beperkingen vertragen de implementatie van IT innovaties?

- Unrealistic expectations and techno-hype
- Difficulty building and modifying IT-based systems
- Difficulty integrating IT-based systems
- Organizational inertia and problems of change
- Genuine difficulty anticipating what will happen

b) Een van die obstakels is dat het moeilijk is om IT systemen te bouwen of te veranderen. Geef aan met welk percentage dat misgaat. Ongeveer 30% van alle IT projecten wordt voortijdig stop gezet. Ongeveer 50% gaat ruim over budget, wordt te laat afgeleverd of voldoet niet aan de vooraf gestelde eisen.

2. Hoofdstuk 12

a) Geef de vijf namen van de levels van het SEI Capability Maturity Model.

b) Een van die vijf is initial. Geef van initial 3 karakteristieken.

antwoord bij a:

initial

repeatably

defined

managed

optimizing

antwoord bij b:

processes are ad hoc

processes are sometimes chaotic

success depends on heroic individual efforts

3. Hoofdstuk 9

a) Geef de 6 meest voorkomende types AI applicatie.

b) Een van die types is intelligent agents. Geef van alle 6 types aan welk soort probleem ze pogen op te lossen. Bij intelligent agents is dat het zelfstandig doorzoeken van een netwerk om nuttige

5. CASE STUDY

Antwoord:

a)

operator error
liability for system failure
inadequate system performance
data error
accidental disclosure of data
hardware malfunction
software bug or design error
damage to physical facilities

zie pag 514 ed IV, fig 13.1

b) operator error. range: 60-80%

zie pag 513 ed IV, "a prime cause" en 60-80 % is on that page too.

c)

IGZ:

operator error: "een verkeerde bloedgroep toegediend"
"ondeugdelijke gebruikt computersysteem"
"fouten gemaakt rond de bloedgroepbepaling en
de uitgifte van bloedproducten"
"niet voldoende gebruik gemaakt
van de mogelijkheden"

AZM:

operator error: "menselijke fouten"
inadequate system performance: "gebrek aan afstemming tussen software en
werkwijze"
"gebrek aan functionaliteit van de software"
software bug or design error: "een probleem in de software"

%%%

Multiple-Choice vragen

6. De uitdaging in het verdelen van het werk tussen mens en machine ligt daarin:

- A) om de machine zoveel mogelijk taken te geven\\
- B) om mensen zoveel mogelijk taken te geven\\
- C) die taken te geven waar ze het beste in zijn\\
- D) geen van de genoemde antwoorden

antwoord: C

7. Mensen die computers aanzien voor gelijken (antropomorfiseren) hebben de neiging om:

- A) de schuld geven aan de computer voor gemakte fouten
- B) computer kwaliteiten toe te kennen aan mensen
- C) vijandigheid voelen naar computers toe
- D) alle genoemde antwoorden

antwoord: A

8. Het vermogen om andere mensen dingen te laten doen heet:

- A) macht
- B) automonie
- C) controle
- D) autoriteit

antwoord: A

9. De vier typen soorten software zijn:

- A) application software, end-user software, system development software, executive software
- B) application software, end-user software, system development software, system software
- C) application software, technical software, system development software, executive software
- D) application software, end-user software, transaction software, executive software

antwoord: B

10. Een manier om werk te verdelen bij het ontwerp van systemen in een verzameling van zelfstandige modules die samenwerken heet:

- A) modulariteit
- B) reusability
- C) procedurality
- d) independence

antwoord: A

11. Het onderhoud van bestaande informatiesystemen het ondersteunen van gebruikers kan opgesplits worden in de volgende categorieën

- A) user support, technical support, bug fixes
- B) user support, system enhancements, technical supports
- C) technical support, system enhancements, bug fixes
- D) user support, system enhancements, bug fixes

antwoord: B

12. Aan het eind van de jaren negentig lag de focus van BPR op:

- A) het verbeteren van processen in plaats van revolutionaire wijzigingen

- B) revolutionaire wijzigingen in plaats van verbeteringen van processen
- C) het verbeteren van het proces en revolutionaire wijzigingen
- D) geen van allen

antwoord: A

13. De vier alternatieve manieren om systemen te ontwikkelen zijn:

- A) traditional system life cycle, prototyping, application packages, "synch and stabilize
- B) object oriented approach, prototyping, application packages, end-user development
- C) traditional system life cycle, prototyping, application packages, end-user development
- D) traditional system life cycle, prototyping, object oriented approach, end-user development

antwoord: C

14. Voor IS-ontwikkelaars lijkt het bouwen van nieuwe systemen vaak dan oude systemen op orde houden

- A) frusterender
- B) moeilijker
- C) uitdagender
- D) onsmakelijker

antwoord: C

15. Wat is de meest voorkomende reden voor het falen van een project in de development fase:

- A) het systeem lijkt te duur
- B) het systeem vraagt een te grote verandering van de bestaande praktijk
- C) de controle op het systeem is niet voldoende
- D) het is te moeilijk om de requirements vast te leggen

antwoord: D

16. Softwarefouten zijn een fundamenteel probleem van gecomputeriseerde systemen, omdat:

- A) je geluk moet hebben om de fouten tegen te komen bij het testen
- B) fouten kunnen zichzelf in een later stadium creëren
- C) er is geen onfeilbare manier om te bewijzen dat programma's correct werken
- D) testers zijn lui

antwoord: C

17. Een verzameling instructies die een gebruiker in staat stelt om langs een computerbeveiliging te komen heet een:

- A) logic bomb
- B) Trojan horse
- C) trap door
- D) virus

antwoord: C

%%%

Practicumvragen

18.

not available as ASCII art; see the werk college
for several examples.

19.

a) The ERM is a high-level conceptual data model. It is meant for the conceptual design of database applications (as opposed to the logical or physical design). Entities are things in the real world. Relationships are conceptual relations between these things in the real world. The ERM does not deal with concerns of query mechanisms, efficient representation and storage. The relational data model provides a logical view on databases. It suggests that a database is collection of tables (or in math-speak relations), where the rows in each table follow a common schema, and the columns are flat values, including the possibility of reference attributes. The relational model is immediately suited for mathematical reasoning based on the relational algebra and for query languages such as SQL.

(This can be further elaborated and explained in different ways.)

b) Here the normalisation steps are needed to derive a suitable relational model from the ERM.

20.

a)

SQL is a language for accessing database systems whereas the relational algebra is mathematical model based on mathematical relations and operations on them. SQL provides technical concepts that are missing in the mathematical model, e.g., a sublanguage for data definition, issues concerned with the physical model such as indexes, special formatting features such as grouping and unique rows. The relational algebra is based on just a few orthogonal operators with a simple semantics whereas SQL is a rather complex language. SQL potentially modifies tables in a database by side effects so to say. The relational algebra is free of side effects. It rather computes relations.

(Several other differences and views are conceivable.)

b)

We suppose a given relation R.

- Selection operation, noted $\sigma_C(R)$, returns only the tuples from R which satisfy the condition c.
- Projection operation, noted $\pi_L(R)$, returns a relation with only the attributes of R specified in the list L.
- Union operation between two relations R and S, noted $R \cup S$, returns all tuples of R and S, with no duplicates.
The relations need to be union compatible, so they need to have the same attributes.
- Intersection between two relations R and S, noted $R \cap S$ returns a relation that contains only the tuples present both in R and in S.
The relations need to be union compatible
- Theta join operation between two relations R and S, noted $R \bowtie_J S$, returns all tuples from $R \times S$ which satisfy the join condition J.

(Examples omitted.)

21.

```
% Id x Name x Age
monteur(m1, 'Klaus', 77).
monteur(m2, 'Rainer', 88).

% Id x Sign x Color
auto(a1, 'ABC111', red).
auto(a2, 'XYZ42', green).
auto(a3, 'QQQ000', white).
auto(a4, 'FOOBAR1', white).

% Id x Name x Auto
eigenaar(e1, 'Hans', a1).
eigenaar(e2, 'Peter', a2).
eigenaar(e3, 'Hendrik', a3).
eigenaar(e3, 'Hendrik', a4).

% Id x Auto x Date
reparatie(r1, a1, 20030102, 'Replaced breaks').
reparatie(r2, a2, 20020527, 'Replaced wheels').
reparatie(r3, a3, 20030115, 'Replaced chassis').
reparatie(r4, a4, 20030116, 'Replaced chassis').

% De naam van de monteurs met een bepaalde leeftijd.
byleeftijd(N, L) :- monteur(_, N, L).

% De naam van de eigenaren die op het
% Buikslotermeerplein wonen en een rode auto in hun bezit hebben.
buik(N) :- eigenaar(_, N, A), auto(A, _, 'red').

% De kleuren van alle autos die in januari 2003 gerepareerd zijn.
januari(C) :- reparatie(_, A, D, _),
              D > 20030100,
```

```
D < 20030200,  
auto(A,_,C).
```

```
% De reparaties (kenteken, datum, omschrijving) van Hendrik  
% Boldersbast's autos.  
hendrik(Sign,Date,Comment) :- eigenaar(_, 'Hendrik', Id),  
                               auto(Id, Sign, _),  
                               reparatie(_, Id, Date, Comment).
```

22.

See the Chapters in "First Steps in Database design".
Name them.
Briefly explain the I/O behaviour for these phases.

23.

a)

data structure extraction
data structure conceptualization.

b)

Implicit structures
Optimized structures
Awkward design
Obsolete structures
Size
Documentation