

Samlede notater for

Datamatikeruddannelsen

1. – 3. semester

Virksomhedsforståelse og systemudvikling

Udgave nr. 1

Skrevet af
Alice Raunsbæk
(alice@m-r-a.dk)

Senest opdateret
Lørdag d. 10. juni 2006

1. SAMLEDE NOTATER FOR DMU (1-3. SEMESTER)

1.1. Forord

1.1.1.1. Opdeling i nogle udgaver

Indenfor uddannelsen er der som sådan fire emner man kan dele fagene op i: virksomhedsforståelse, systemudvikling, programmering og netværk. Grundet størrelsen af dokumentet findes disse typisk i forskellige dokumenter – du kan på forsiden (og forhåbentlig på filens navn) se hvilken udgivelse du har fat i.

Typisk forefindes udgivelser kun i PDF. Ønskes de i et andet format kontakt da redaktøren.

1.1.1.2. Tegn og tekst forklaring

Forskellige ting har forskellige formateringer.

Hvis du ser tekst af denne type er der tale om java-kode (eller kode i det hele taget).

For forståelsens skyld er der iblandt notaterne også opgaveløsninger. Dem kan du kende på den grå farve denne tekst også har.

Over den grå tekst eller inden i den ser du måske noget med denne lysere farve – det er opgave formuleringer.

Forklaringer

Når noget skal uddybes så har det typisk en normal linje (eller en med fed) hvorefter der er en indrykket tekst – det er forklaringen på/beskrivelsen af linjen.

1.1.1.3. Opbygning

Da der er en del notater og dette er et meget langt dokument har jeg valgt at have to indholdsfortegnelser, en overordnet – så du lettere kan finde et fag eller et emne. Den indeholder kun to niveauer: Fag og Emner.

Den mere detaljerede indeholder alle niveauer (der er ned til 4. niveau).

1.1.1.4. Forbehold

Disse notater er skrevet på baggrund af min egen forståelse af hvad lærerne har forkyndt i timerne. Jeg kan derfor ikke garantere for at de er 100% korrekte.

Der kan endvidere være 'huller' i notaterne, dette kan enten skyldes at jeg synes at det virkede himmelråbende logisk da jeg skrev notaterne eller at jeg fattede hat af emnet.

Finder du ellers fejl (eller mangler) er du velkommen til at skrive til mig med rettelser.

Mvh. Alice Raunsbæk

alice@m-r-a.dk

1.2. Overordnet indholdsfortegnelse

| | | |
|-------|--|----|
| 1. | SAMLEDE NOTATER FOR DMU (1-3. SEMESTER) | 2 |
| 1.1. | Forord | 2 |
| 1.2. | Overordnet indholdsfortegnelse | 3 |
| 1.3. | Detaljeret indholdsfortegnelse | 5 |
| 2. | VIRKSOMHEDSFORSTÅELSE | 9 |
| 3. | ITO, 1. SEMESTER | 9 |
| 3.1. | Individets adfærd styres af flere variabler (s. 115) | 9 |
| 3.2. | Grupper (kap 7) | 11 |
| 3.3. | Ledelse | 11 |
| 3.4. | Organisation | 12 |
| 3.5. | IT | 16 |
| 3.6. | Analyse | 16 |
| 4. | ITO, 2. SEMESTER | 17 |
| 4.1. | E-business | 17 |
| 4.2. | Økonomi | 17 |
| 4.3. | Enterprise Resource Planning (ERP) | 18 |
| 4.4. | Rige billeder | 19 |
| 4.5. | Effektvurdering af edb-investeringer | 19 |
| 4.6. | Standardsystemer | 19 |
| 4.7. | IT sikkerhed | 21 |
| 5. | SYSTEMUDVIKLING | 22 |
| 6. | SD, 1. SEMESTER | 22 |
| 6.1. | Introduktion | 22 |
| 6.2. | Udviklingsmetoder | 24 |
| 7. | SD, 2. SEMESTER | 25 |
| 7.2. | Systemudviklings niveauer | 25 |
| 7.3. | Faserne i UP | 26 |
| 7.4. | Dokumentation | 28 |
| 7.5. | Eksperimenter | 28 |
| 7.6. | Projektarbejde | 29 |
| 8. | SUM, 3. SEMESTER | 30 |
| 8.2. | Kvalitet i systemudvikling | 30 |
| 8.3. | Produkt og proces i systemudviklingen | 31 |
| 8.4. | Kravspecifikation | 32 |
| 8.5. | Planlægning og styring af test | 34 |
| 8.6. | V-modellen | 36 |
| 8.7. | Forundersøgelse (FU) | 37 |
| 8.8. | MUST-metoden | 40 |
| 8.9. | Brugerinddragelse i systemudvikling | 45 |
| 8.10. | Kvalitetsindbygning | 47 |
| 8.11. | Fysisk design (del af interaktionsdesign) | 50 |

| | | |
|-------|------------------------------------|----|
| 8.12. | Mønstre i analyse..... | 51 |
| 8.13. | Design af (logisk) arkitektur..... | 51 |
| 8.14. | Vedligeholdelse | 53 |
| 8.15. | Implementering..... | 55 |
| 8.16. | Procesforbedring..... | 57 |

1.3. Detaljeret indholdsfortegnelse

| | |
|--|----|
| 1. SAMLEDE NOTATER FOR DMU (1-3. SEMESTER) | 2 |
| 1.1. Forord | 2 |
| 1.1.1.1. Opdeling i nogle udgaver | 2 |
| 1.1.1.2. Tegn og tekst forklaring | 2 |
| 1.1.1.3. Opbygning | 2 |
| 1.1.1.4. Forbehold | 2 |
| 1.2. Overordnet indholdsfortegnelse | 3 |
| 1.3. Detaljeret indholdsfortegnelse | 5 |
| 2. VIRKSOMHEDSFORSTÅELSE | 9 |
| 3. ITO, 1. SEMESTER | 9 |
| 3.1. Individets adfærd styres af flere variabler (s. 115) | 9 |
| 3.1.1.1. Kommunikation (s. 121) | 9 |
| 3.1.1.2. Perception/opfattelse | 9 |
| 3.1.2. Motivation | 10 |
| 3.1.2.1. Marslows behovsteori (s. 130) | 10 |
| 3.1.2.2. Herzbergs hovedfaktorer (s. 136) | 10 |
| 3.1.2.3. Vrooms/Lawlers forventningsteori (s. 143) | 10 |
| 3.1.2.4. Målsættelsesteorier | 10 |
| 3.1.3. Opgave - Sentinel | 10 |
| 3.2. Grupper (kap 7) | 11 |
| 3.2.1.1. Typer af grupper | 11 |
| 3.3. Ledelse | 11 |
| 3.3.1.1. Elementer af begrebet ledelse (s. 229) | 11 |
| 3.3.1.2. Mintzberg: god ledelse (s. 241) | 11 |
| 3.3.1.3. Adizes lederroller (s. 236) | 12 |
| 3.4. Organisation | 12 |
| 3.4.1.1. Indhold | 12 |
| 3.4.1.2. Weber: Den bureaukratiske skole 1864-1920 Sociolog (s. 39) | 12 |
| 3.4.1.3. Taylor: Den videnskabelige skole 1856-1915 Ingeniør (s. 41) | 12 |
| 3.4.1.4. Fayol: Den administrative skole 1841-1925 (s. 44) | 13 |
| 3.4.1.5. Sammenfatning | 13 |
| 3.4.1.6. Arbejdsdeling | 13 |
| 3.4.2. Historisk for organisationer | 13 |
| 3.4.2.1. Nyere organisationsformer | 13 |
| 3.4.2.2. Projektorganisation (s. 80-84) | 13 |
| 3.4.2.3. Matrixorganisation (s. 85-87) | 14 |
| 3.4.2.4. Opgave - Revisionsfirmaet | 14 |
| 3.4.2.5. (Delvis) Selvstyrende grupper | 14 |
| 3.4.3. Organisationsdesign | 14 |
| 3.4.3.1. Mekanisk og Organisk organisation | 15 |
| 3.5. IT | 16 |
| 3.5.1. Informationssystem (IS) | 16 |
| 3.5.1.1. Formål og funktion | 16 |
| 3.5.2. IT som virkemiddel | 16 |
| 3.6. Analyse | 16 |
| 4. ITO, 2. SEMESTER | 17 |
| 4.1. E-business | 17 |
| 4.1.1.1. Forretningsmodeller | 17 |

| | | |
|----------|---|----|
| 4.1.1.2. | Indtjeningsmodeller | 17 |
| 4.2. | Økonomi | 17 |
| 4.2.1. | Regnskaber | 18 |
| 4.3. | Enterprise Resource Planning (ERP) | 18 |
| 4.3.1. | Fire lag | 18 |
| 4.3.1.1. | Udvikling | 18 |
| 4.3.1.2. | Fordele | 18 |
| 4.3.1.3. | Ulemper | 19 |
| 4.4. | Rige billeder | 19 |
| 4.5. | Effektvurdering af edb-investeringer | 19 |
| 4.5.1.1. | Cost-benefit betragtning | 19 |
| 4.5.1.2. | Visions- og vidensgenerator | 19 |
| 4.6. | Standardsystemer | 19 |
| 4.6.1.1. | Typer | 19 |
| 4.6.1.2. | Udvikling/udvælgelse af nye systemer | 20 |
| 4.6.2. | Standardsystemer I Virksomheder (SIV) | 20 |
| 4.7. | IT sikkerhed | 21 |
| 4.7.1.1. | Trusler | 21 |
| 4.7.2. | Hvorfor er IT-sikkerhed svær? | 21 |
| 4.7.3. | Handlingsplan | 21 |
| 4.7.4. | Løsningskategorier | 21 |
| 5. | SYSTEMUDVIKLING | 22 |
| 6. | SD, 1. SEMESTER | 22 |
| 6.1. | Introduktion | 22 |
| 6.1.1.1. | Enumeration = opremsning | 22 |
| 6.1.1.2. | Tilgange | 23 |
| 6.1.1.3. | Association | 23 |
| 6.1.1.4. | Aggregation | 23 |
| 6.1.1.5. | Komposition | 23 |
| 6.1.1.6. | Abstrakte klasser | 24 |
| 6.2. | Udviklingsmetoder | 24 |
| 6.2.1. | Rumbaugh – udviklingstrin | 24 |
| 6.2.2. | Udviklingslivscyklus | 24 |
| 6.2.2.1. | Hvor kommer det nye system fra | 24 |
| 6.2.3. | Hvad er en god systemdefinition | 25 |
| 7. | SD, 2. SEMESTER | 25 |
| 7.1.1.1. | Arlow-bog | 25 |
| 7.2. | Systemudviklings niveauer | 25 |
| 7.2.1.1. | Aksiomer (grundantagelser) | 25 |
| 7.2.1.2. | Risici i Kajs biler | 26 |
| 7.3. | Faserne i UP | 26 |
| 7.3.1.1. | Inception (resultat/mål) | 26 |
| 7.3.1.2. | Elaboration (resultat/mål) | 27 |
| 7.3.1.3. | Construction (resultat/mål) | 27 |
| 7.3.1.4. | Transition (resultat/mål) | 28 |
| 7.4. | Dokumentation | 28 |
| 7.5. | Eksperimenter | 28 |
| 7.6. | Projektarbejde | 29 |
| 7.6.1.1. | Definition: udforskende læring | 29 |
| 7.6.2. | Projektarbejde på 2. semester | 29 |

| | | |
|----------|---|----|
| 7.6.2.1. | Mål | 29 |
| 7.6.2.2. | Samarbejde | 29 |
| 7.6.2.3. | Lærernes rolle | 30 |
| 8. | SUM, 3. SEMESTER | 30 |
| 8.1.1. | Fagets indhold | 30 |
| 8.2. | Kvalitet i systemudvikling..... | 30 |
| 8.2.1. | Opgave 1 | 31 |
| 8.3. | Produkt og proces i systemudviklingen | 31 |
| 8.3.1. | Opgave 2 | 32 |
| 8.4. | Kravspecifikation..... | 32 |
| 8.4.1. | Hvad er det? | 32 |
| 8.4.2. | Definition | 33 |
| 8.4.3. | Indhold | 33 |
| 8.4.4. | Ikke-funktionelle krav | 33 |
| 8.4.5. | Kvalitet af kravspecifikation | 34 |
| 8.4.6. | Kvalitetsindbygning i kravspecifikationer | 34 |
| 8.4.7. | Check af kravspecifikationens kvalitet | 34 |
| 8.5. | Planlægning og styring af test..... | 34 |
| 8.5.1. | Planlægningsniveauer | 34 |
| 8.5.2. | Testspecifikation | 35 |
| 8.6. | V-modellen..... | 36 |
| 8.6.1. | Opgave 3 fra dag 2 | 36 |
| 8.7. | Forundersøgelse (FU) | 37 |
| 8.7.1. | Definition | 37 |
| 8.7.2. | Resultat af forundersøgelse | 39 |
| 8.7.3. | Viderearbejde | 39 |
| 8.8. | MUST-metoden | 40 |
| 8.8.1. | Principper i MUST | 40 |
| 8.8.1.1. | Opgave 5 fra sidst | 41 |
| 8.8.2. | Forberedelse | 41 |
| 8.8.3. | Fokusering: Strategi analyse..... | 42 |
| 8.8.4. | Fordybelse: Analyse af arbejdsgange | 42 |
| 8.8.5. | Fornyelse | 42 |
| 8.8.6. | Opgaveløsning med brug af MUST | 43 |
| 8.8.6.1. | Opg. 1, Forberedelsesfasen | 44 |
| 8.8.6.2. | Opg. 2 Fokuseringsfasen..... | 44 |
| 8.8.6.3. | Opg. 3 Fordybelsesfasen..... | 45 |
| 8.8.6.4. | Opg. 4 Indvendinger (imod ideer) | 45 |
| 8.9. | Brugerinddragelse i systemudvikling..... | 45 |
| 8.9.1. | Problemer med opbakning (generelt) | 45 |
| 8.9.1.1. | Indvendinger (imod ideer) | 45 |
| 8.9.1.2. | Mulige modargumenter..... | 46 |
| 8.9.2. | Modstand mod forandring | 46 |
| 8.9.3. | Brugerdeltagelses niveauer..... | 46 |
| 8.9.4. | Områder med mulighed for aktiv deltagelse | 47 |
| 8.10. | Kvalitetsindbygning..... | 47 |
| 8.10.1. | Måder | 47 |
| 8.10.2. | Mønstre | 48 |
| 8.10.3. | Analysemønstre | 48 |
| 8.10.4. | Mønster-mønster (!)..... | 48 |
| 8.10.5. | Brugergrænsefladedesign..... | 48 |
| 8.10.6. | Molic (repetition)..... | 48 |

| | | |
|-----------|--|----|
| 8.10.7. | Metode (Benyon, Turner & Turner) | 48 |
| 8.10.8. | Begrebsmæssigt design | 49 |
| 8.10.9. | Mønstre i interaktion (eksempler)..... | 50 |
| 8.11. | Fysisk design (del af interaktionsdesign) | 50 |
| 8.12. | Mønstre i analyse..... | 51 |
| 8.12.1. | Use-case specifikation | 51 |
| 8.12.2. | Use-case sammenhænge | 51 |
| 8.13. | Design af (logisk) arkitektur | 51 |
| 8.13.1. | Logisk arkitektur | 51 |
| 8.13.2. | Arkitekturanalyse | 52 |
| 8.13.3. | Basale principper/teknikker | 52 |
| 8.13.4. | Basal løsning | 52 |
| 8.13.5. | UP og arkitektur..... | 53 |
| 8.14. | Vedligeholdelse | 53 |
| 8.14.1. | Typer af vedligeholdelse..... | 53 |
| 8.14.2. | Reduktion af omkostninger..... | 54 |
| 8.14.3. | Organisering..... | 54 |
| 8.14.4. | Standardforløb for vedligeholdelse | 54 |
| 8.14.5. | Reverse engineering | 54 |
| 8.15. | Implementering | 55 |
| 8.15.1. | Overordnet..... | 55 |
| 8.15.2. | Komponenter..... | 55 |
| 8.15.3. | Implementationsplan | 56 |
| 8.15.3.1. | Faser og faselinier..... | 56 |
| 8.16. | Procesforbedring | 57 |
| 8.16.1. | Intro..... | 57 |
| 8.16.2. | CMMI (Capability Maturity Model Integration) | 57 |
| 8.16.2.1. | Består af fem niveauer | 57 |
| 8.16.3. | Opsamling..... | 58 |

2. VIRKSOMHEDSFORSTÅELSE

Virksomhedsforståelse dækker over Informations Teknologi i Organisationen (ITO).

3. ITO, 1. SEMESTER

3.1. Individets adfærd styres af flere variabler (s. 115)

Personlighed:

- Perception: opfattelse
 - Motivation
 - Engagement
 - Trivsel
 - Job udformning
- Individuel adfærd

3.1.1.1. Kommunikation (s. 121)

Formel:

1. give en forståelse for medarbejderens situation, opgaver og problemer
2. gode ideer
3. ros
4. gode relationer på tværs

Kommunikere: at gøre fælles (latin)

1. verbal kommunikation: det, der faktisk siges
2. ekstra-verbal kommunikation: måden det siges på
3. non-verbal kommunikation: kropslige signaler

Forstyrret/mislykket kommunikation

1. regulær støj, budskabets udformning (medier)
2. selektiv fortolkning, forventninger, følelsesmæssig tilstand, virkelighedsforståelse
3. sprogbeherskelse, syntaks (sætningsopbygning), semantik (betydning og indhold)

3.1.1.2. Perception/opfattelse

Forældre-tilstand (overjeg) (s. 125)

- kritiske budskaber kan være dømmende, moraliserende, kontrollerende etc.
- Hjælpsomme budskaber kan være beskyttende, trøstende, opmuntrende etc.

Voksen-tilstand (jeg) (s. 126)

- observerer, indsamler objektive informationer, vurderer konsekvenser, træffer valg uden følelsesmæssige aspekter etc.

Barne-tilstand (id) (s. 125)

- det naturlige barn: spontan, nysgerrig, impulsiv, stædig etc.
- det tilpassede barn: oprørsk, underkastende, angst, surmulende etc.

- den lille professor: intuitiv, kreativ, manipulerende etc.

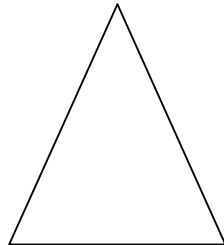
Transaktioner mellem de forskellige tilstande (s. 126-127)

3.1.2. Motivation

Villigheden til at yde en indsats.

3.1.2.1. Marslows behovsteori (s. 130)

- Selvrealisering
- Ego behov
- Sociale behov
- Tryghed
- Fysiske behov



3.1.2.2. Herzbergs hovedfaktorer (s. 136)

Vedligeholdelsesfaktorer

Er med til at stabiliserer opfattelsen af arbejdspladsen så der ikke opstår utilfredshed.

Motivationsfaktorer

Vedrører selve arbejdets indhold, er i stand til at skabe tilfredshed.

- løn
- glæden ved arbejdet
- interesseret – spændende opgaver/muligheder
- fleksibilitet – fritid og frihed
- kolleger – respekt og socialt
- intellektuel udfordring
- personlig udvikling, lære nyt
- at kunne se resultatet (jobrotation, på eget niveau), (jobberigelse, på højere niveau, planlægning)

3.1.2.3. Vrooms/Lawlers forventningsteori (s. 143)

+ Kan indsats føre til en Præstation? (I → P)

+ Kan præstationen føre til en belønning? (P → B)

+ Er værdien tilstrækkelig stor? (V)

= Motivationen er tilstede

Behov -> værdi, forventninger -> mål -> handling

3.1.2.4. Målsættelsesteorier

1. sæt høje men opnåelige mål
2. sæt klare mål (bl.a. tidshorisont)
3. få mål accept (at de der skal udføre det også er enige med det)

3.1.3. Opgave - Sentinel

1. Beskriv kort caseforløbet

2. Hvilke problemer knytter sig til forløbet
3. Hvad er årsagerne til problemer
4. Hvordan kan vi komme videre

3.2. Grupper (kap 7)

En gruppe er en samling mennesker, der:

- har samme mål
- gensidigt påvirker hinanden
- føler samhørighed
- opfatter sig selv som en gruppe

I en gruppe er der normalt roller, normer (formelle såvel som uformelle) og hvert medlem har en form for status.

Synergieffekt

Medlemmernes samlede præstation er større end medlemmernes individuelle præstationer lagt sammen.

3.2.1.1. Typer af grupper

- Formel
 - o Fx salgsafdelingen, fodboldholdet
- Uformel
 - o Fx fælles interesser
- Primær
 - o Tæt og hyppig interaktion (3-6 personer)
- Sekundær
 - o Mindre hyppig og upersonlig interaktion

3.3. Ledelse

Ledelse består af et samspil mellem mennesker, for at nå et fælles mål ved hjælp af fælles midler.

3.3.1.1. Elementer af begrebet ledelse (s. 229)

Management:

= faglig kompetence, administrative egenskaber

Leadership:

= personlige egenskaber, karisma, visioner, dialog

Coaching:

= motivation, udvikle medarbejderen

3.3.1.2. Mintzberg: god ledelse (s. 241)

1. Velinformeret, også på gulvet
2. Oprigtighed/6 år egne veje

3. Eftertænsom, forståelse af branchen og virksomheden

Ledertyper:

Den beregnende: tal besat.

Den heroiske: tror han/hun er kunstner/kreativ vidunderlig

Den engagerede: dybt kendskab til branche, virksomhed og folk.

3.3.1.3. Adizes lederroller (s. 236)

P. Producent:

= handlings- og resultatorienteret.

A. Administrator:

= strategi og mål fastsættelse. Koordinering af funktioner systematisk.

E. Entrepreneur:

= kreativ og igangsætter.

I. Integrator:

= sikrer samarbejde. Ser helheden.

3.4. Organisation

Er en gruppe af mennesker, der arbejder sammen om at opnå fælles mål.

En gruppe består af

- mennesker
- forholdet organisationer imellem
- struktur

3.4.1.1. Indhold

1. Hvem opstiller mål og strategier for virksomhedens opgaver.
2. Hvordan fordeles opgaver på medarbejdere.
3. Hvem skal have ansvaret for opgavernes udførelse.
4. Efter hvilke retningslinier skal opgavernes udføres.

3.4.1.2. Weber: Den bureaukratiske skole 1864-1920 Sociolog (s. 39)

- En leder
- En administrativ stab
- Hierarki på 2-3 niveauer

3.4.1.3. Taylor: Den videnskabelige skole 1856-1915 Ingeniør (s. 41)

(Den videnskabelige skole, Scientific Management)

- Arbejdsdeling på individniveau
- Specialisering
- Kontrol
- Belønningssystem

Forløber for samlebandet.

3.4.1.4. Fayol: Den administrative skole 1841-1925 (s. 44)

- Koordinering mellem de 6 funktioner (s. 45)
 - o Tekniske (produktionsprocesser, vedligeholdelse af maskiner)
 - o Handelsmæssige (salg og indkøb)
 - o Finansielle (kapital fremskaffelse)
 - o Sikkerhedsmæssige (beskytte individer, materiel og miljø)
 - o Regnskabsmæssige (regnskaber, kalkulationer, bogholderi)
 - o Administrative (planlægning, koordinering og kontrol)

3.4.1.5. Sammenfatning

- Klar adskillelse at planlægge og udførelse
- Medarbejdernes tro på lederens autoritet
- Lederen er kreativ
- Medarbejderen motiveres gennem penge
- Organisationen er en velsmurt maskine

3.4.1.6. Arbejdsdeling

Horisontalt:

- Objektprincippet (ex. TV, CD..)
- Funktionsprincippet (ex. Salg, indkøb..)

3.4.2. Historisk for organisationer

| | | |
|----------------------------------|----|------------------|
| I gamle dage | -> | Nyere tid |
| Stabile miljø/omgivelser, kunder | -> | dynamisk |
| Produktionstankegang | | |
| - kvantitet | | |
| - - omkostningsminimering | -> | markedstankegang |

3.4.2.1. Nyere organisationsformer

- Sammenhæng mellem trivsel og produktivitet
- Motivation
- Medarbejdere er bedre uddannet
- Nye opgavetyper (ex. Planlægning og kontrol, ikke alene udførelse)
- Tværfunktionelle opgaver

3.4.2.2. Projektorganisation (s. 80-84)

- Løser enkeltstående opgaver og opløses derefter
- Opgave er tids- og ressourcekrævende
- Opgave af anderledes karakter
- Består af en styregruppe og en projektgruppe

Styregruppe

- sætter rammerne for projektet

Projektgruppe

- gennemfører arbejdet

Fordele

- Udvikler virksomhedens forandringsevne
- Sikrer anvendelse af tværfaglig viden
- Udnytter specialist viden
- Udvikler medarbejderne fagligt og personligt
- Skaber engagement i virksomheden

Ulemper

- Stresset arbejdssituation
- Rollekonflikter mellem to chefer
- Vanskeligt at vende tilbage til basisorganisation

3.4.2.3. Matrixorganisation (s. 85-87)

- Permanent supplement til basisorganisationen
- Gentagende projektopgaver
- Afdelingsleder har fagligt ansvar (med til at udvælge personer til projektet)
- Projektleder har ansvar overfor kunden (deadlines m.v.)

Fordele

- Fleksibel organisationsform
- Udnytter ekspertviden
- Fremmer kreativitet
- Jobberigende

Ulemper

- Medarbejdere har flere tilhørsforhold
- Kræver koordinering
- Konflikter mellem afdeling og projektledelse

3.4.2.4. Opgave - Revisionsfirmaet

Opgavetype: ~~standard~~/kompleks

3.4.2.5. (Delvis) Selvstyrende grupper

Hensigt: Synergi i gruppearbejde.

Hovedtanker: Motivation -> effektivitet

Opgaver løses i gruppen

Gruppen træffer selv beslutninger om opgaveløsningen

Grad af selvstyre: (s. 90)

3.4.3. Organisationsdesign

Situationsbetingede teorier

Galbraith: Information

Nødvendig information

- Tilgængelig information

= Opgaveusikkerhed

Bestemmes af:

1. Opgavens natur
2. Præstationsniveauet

Kan løses ved:

- Reduktion af informationsbehandlingsbehovet
- 1. Ekstern tilpasning -> tættere samarbejde/opkøbe leverandør
- 2. Etablering af ekstra ressourcer
 - Indbygge frihedsgrader
 - færre vare- og kundegrupper
 - lager
 - leveringstid= Nedsætte præstationsniveauet
- 3. Selvstyre
 - objektopdeling
 - divisionalisering
- Øger evnen til at behandle information
- 4. Investering i vertikale informationssystemer (bedre styring)
 - styringsprincipper (JIT – Just In Time, produktionsplanlægning, ERP, databaser)
- 5. Etablere tværgående koordinering
 - uformelle kontakter
 - møder
 - koordination

3.4.3.1. Mekanisk og Organisk organisation

Mekanisk

- Rutiner
- Masseproduktion
- Standardløsninger
- Stabile omgivelser / forudsigelighed
- Enkel / programmerbar
- Homogent / ingen præference
- Lav fjendtlighed / lille konkurrence

Organisk

- Kreativitet
- Stykordre produktion
- Individuelle løsninger
- Dynamiske omgivelser / uforudsigelighed

- Komplex / ikke så programmerbar
- Heterogent / præferencer
- Høj fjendtlighed / stor konkurrence

Omgivelser

Jo mere man ved om hvad kunderne vil jo mere stabilt.

Stabilt: at man f.eks. altid spiser samme mængde kartofler og mælk.

Stabilt: ting der er råvare baseret.

3.5. IT

Data

= formaliseret repræsentation af symboler og signaler, som er egnet til bearbejdning.

Information

= data som er fortolket i forhold til et fysisk objekt eller begreb

Viden

= målrettet anvendelse af information

Informationsteknologi

= videns- og erfaringsområde der knytter sig til udvikling af informationssystemer

- forståelse eller håndtering af teknisk udstyr (PC, netværk)

3.5.1. Informationssystem (IS)

= håndterer information om objekter, som man interesserer sig for

3.5.1.1. Formål og funktion

Styringsformål: Frembringe information som kan bruges til planlægning og beslutningstagning

Kommunikationsformål: Skabe kontakt mellem forskellige dele af virksomheden

Læringsformål: Øge medarbejdernes indsigt og kundskaber

-> opfatte, bearbejde og sprede information

3.5.2. IT som virkemiddel

- 1- Rationaliseringsværktøj: gøre ting rigtigt (billigt)
- 2- Måleffektivisering: gøre de rigtige ting (bedre)
 - beslutningsstøttesystemer
- 3- Konkurrenceeffektivitet: markedsorientering
 - i. Udvikling af nye produkter og tjenester
 - ii. Øget nærhed
 - iii. Samspil mellem virksomheder
 - iv. Reduktion af produktions- og distributionsomkostninger (optimering)

3.6. Analyse

SWOT – for at den er besvaret ordentligt skal man kunne svare på:

Hvad: branche, produkter, behov

Hvor: geografisk område

Hvem: målgruppe

Hvordan: virksomhedstype

4. ITO, 2. SEMESTER

4.1. E-business

Er en transformation af virksomhedens værdikædeaktiviteter ved anvendelse af Internet og andre digitale teknologier.

4.1.1.1. Forretningsmodeller

- 1) E-shops – webbutikker, webmarketing
- 2) E-malls – e-supermarkeder, markedspladser
- 3) E-auctions
E-procurement (køberstyret), udbudsstyring (gatetrade.net)
E-auktion (sælgerstyret), en virksomhed eller private sætter varer til salg
- 4) Virtual Communities – bygger på fællesinteresser
- 5) Information Brokage – informationsudbydere (mybanker.dk)

4.1.1.2. Indtjeningsmodeller

- 1) Direkte salg af produkter eller services
- 2) Abonnering eller leje af services
- 3) Kommissionsbaseret salg – henvisning, auktion, mellemandlersalg
- 4) Markedsføring

4.2. Økonomi

Drejer sig om anvendelsen af knappe ressourcer.

Økonomistyring

Træffe og gennemfør rationelle handlinger.
Måleenhed: kr.

Regnskab

Økonomisk beskrivelse af handlinger i fortiden

Budget

Økonomisk beskrivelse af handlinger i en fremtidig periode

Omsætning

Mængden * prisen

Variable omkostninger

Variere direkte ved salg og produktionens størrelse

Faste omkostninger/kapacitets omk.

Omkostninger som må afholdes uanset om produktion og salg er stor eller lille

4.2.1. Regnskaber

Eksternt regnskab

Til folk udenfor virksomheden. Der er regler for udformningen.

Told & skat, leverandører, investorer, banker

Internt regnskab

Typisk mere detaljeret end de eksterne regnskaber

Årsrapport/Årsregnskab

Formål: At give relevant og pålidelig information om virksomhedens økonomiske situation, og samtidig give grundlag for virksomhedens evne til at tjene penge.

Ledelsesberetning og ledelsespåtegning

Resultatopgørelse med noter, Indtægt - omkostninger = resultat

Balance med noter, øjebliksbillede

Pengestrømme

4.3. Enterprise Resource Planning (ERP)

Filosofi

Styre samtlige virksomhedens ressourcer i et system. Materiale, produktionsapparat, personale, kapital.

Logistisk tankegang (at logistikken skal fungere bedre og mere effektivt)

Standardsystem (mest kendte er SAP, Oracle, herefter Axapta (MS))

Modulopbygget, dvs. det er muligt at til- og fravælge moduler afhængig af behov

Procesorienteret

4.3.1. Fire lag

Basale regnskabsmoduler (finans, budget, likviditet)

Restkontomoduler (debitor, kreditor lager)

Operationelle moduler (materiale- og produktionsstyring)

Kommunikationsmoduler (EDI, Internet)

4.3.1.1. Udvikling

70: automatisering af fx faktura

80: standardøkonomi + logistik

90: fuldt integreret system → realtid

2000: fokus på CRM

4.3.1.2. Fordele

- logistisk effektivitet
- datakonsistens
- gennemsigtighed i logistikprocesser
- best practice

4.3.1.3. Ulemper

- virksomheden må tilpasses systemet
- implementering af systemet er meget svær og omfattende
- økonomisk stort projekt

4.4. Rige billeder

Startproblem (arbejdsgange, forretningsgange, produkt, kunder, brugere, relationer, strategi, mål)

4.5. Effektivvurdering af edb-investeringer

4.5.1.1. Cost-benefit betragtning

- Investeringsomkostninger
- Besparelser
- Merindtægter

4.5.1.2. Visions- og vidensgenerator

- 1) Direkte og umiddelbare effekter. Typisk økonomiske cost-benefit betragtninger.
- 2) Indirekte og funktionsafhængige effekter
 - o Ændringer i eksisterende kendte funktioner
 - Nye forretningsgange ex. sagsbehandlingssystem → tværgående sagsbehandling
- 3) Langsigtede og strukturafhængige effekter. Mere grundlæggende organisatoriske omlægninger, i f.eks. ansvar og kompetencefordeling.
- 4) Strategiske og omverdensbetingede effekter. Forandringer af omverdenens arbejdsvilkår
 - a. Tættere kommunikations nærværk → logistik

4.6. Standardsystemer

IT-system der er udviklet til brug i mange virksomheder.

Hvem udvikler standardsystemer?

- Maskinleverandør
- Tredjepartsleverandører (software firma der kun laver den type system)
- Brugervirksomhed

4.6.1.1. Typer

1. Udviklingsansvar
2. Omfang
Totalsystem contra specialområde
3. Ydeevne
4. Styringsfilosofi (centralisering | decentralisering)
5. Tilpasningsmuligheder
 - a. Helt standardiseret = ingen tilpasningsmuligheder (fx operativsystemer)

- b. Hårdt standardiseret system = få tilpasningsmuligheder (fx ERP-systemer, økonomisystemer, regelbaserede opgaver)
- c. Standardiseret system = tilpasning mulig (fx kodetilpasning ERP-system)
- d. Let standardiseret system = tilpasning sker med grundlag i eget system (fx rammesystem, Excel)

6. Tilpasningsmåden

- a. Hårdkodet – kompliceret at lave ændringer i programkoden
- b. Tabelstyret – fleksibilitet i modulvalg
- c. Programmerbart – rammeværk fx regneark

4.6.1.2. Udvikling/udvælgelse af nye systemer

Traditionel: foranalyse + analyse + design + realisering + implementering

Standardsystem: foranalyse + valg + tilpasninger + implementering

4.6.2. Standardsystemer I Virksomheder (SIV)

0. Behovsanalyse. Grundig foranalyse

- a. Kravspecifikation
- b. Cost-benefit analyse

1. Forudsætningsanalyse

- a. Yderligere krav
- b. Prioritering af krav

2. Markedsundersøgelse

- a. Indsamling af informationer om standardsystemer og leverandører

3. Leverandørvurdering

- a. Vurdering af
 - i. Udbud
 - ii. Ansvarlighed
 - iii. Kvalitet
 - iv. Økonomi

4. Indhentning af tilbud

- a. Omkostning ved kød af system
 - i. Licitation (at flere får lov at byde ind)

5. Sammenligning

- a. Sammenlign systemer systematisk
- b. Krav

6. Sortering

- a. Finsortering → 2-3 bedste
- b. Systemer udvælges

7. Demonstration

- a. Demo hos leverandører eller brugere af systemet
- 8. Behovskomplettering
 - a. Revurdering af kravene fra pkt. 0 og 1
- 9. Sammenligning og slutvurdering
 - a. Førstevalget udsættes for detaljeret check
- 10. Præliminærvalg
 - a. Foreløbigt valg til testkørsel
- 11. Testkørsel
 - a. Helst hos en selv
- 12. Forhandling
- 13. Beslutning og aftale
 - a. Kontrakt indgås
- 14. Besked til andre leverandører
 - a. Give besked til andre leverandører om valget

4.7. IT sikkerhed

IT sikkerhed modsvarer almindelig sikkerhed:

- vira versus sygdom
- Trojanske heste versus ubudne gæster

4.7.1.1. Trusler

Der er flere forskellige typer af trusler i forhold til IT-sikkerheden: personrelaterede trusler, systemmæssige trusler, trusler udenfor virksomhedens kontrol.

4.7.2. Hvorfor er IT-sikkerhed svær?

1. Sikkerhed er en barriere for fremskridt
2. Sikkerhed tillært adfærd

4.7.3. Handlingsplan

1. Virksomhedskendskab - eksisterende foranstaltninger og proceskendskab
2. Kendskab til mulige løsninger og deres effekt

4.7.4. Løsningskategorier

1. Adgangskontrol og funktionsadskillelse
2. Personafhængighed
3. Antivirus
4. Internetadgang
5. Hjemmearbejdspladser
6. Fysisk beredskab

5. SYSTEMUDVIKLING

Systemudvikling dækker over Software Design (SD) og System Udviklings Metode (SUM).

6. SD, 1. SEMESTER

6.1. Introduktion

Objekt Orienteret Modellering – overblik, model der kan diskuteres ud fra, undgå fejl under implementering.

Klassediagram – viser klasser, hvad de indeholder samt hvilke forbindelser der er imellem dem. Overordnet – mulige sammenhænge imellem data.

Analyse – hvad skal vi arbejde med (hvilke klasser)

Logiske valg, associeringer, mangfoldigheder, ingen linkattributter.

Design – hvordan skal det gøres (hvilken database)

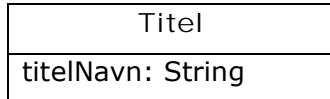
Korrekte valg også ud fra programmeringssprog, retninger på associationer, linkattributter, kan også indeholde forskellige vinduesklasser i GUIen.

6.1.1.1. Enumeration = opremsning.

Card = spar, hjerter, klør, ruder.

Skak = im, fm, gm, wim, wgm.

1. som selvstændig klasse



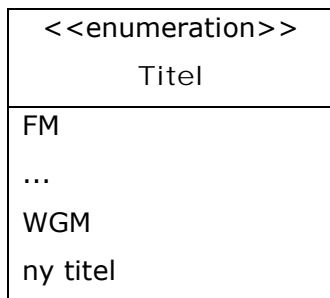
```
Titel fm = new Titel("FM");  
...  
Titel wgm = new Titel("WGM");
```

Sættes I en ArrayList, "Titler" indeholdende alle Titel-objekter. // ArrayList kan variere i størrelse.

Fordel: der kan tilføjes titler uden at kode.

Ulempe: kompleks løsning og kræver initialisering.

2. som enumeration



```
String[] titler = new String[] {"FM", "IM", .., "WGM", ny titel}; // Array har fast størrelse.
```

Fordel: hurtigt og kræver ikke initialisering.

Ulempe: ny titel kræver kodning.

6.1.1.2. Tilgange

+ public, kan bruges i klassen, pakken og ude fra

- private, kan kun bruges i klassen

protected

~ package kan kun bruges i pakken

Klasser er normalt public. Attributter er normalt private. Operationer er mest public.

6.1.1.3. Association

Binær []-[]

Tertier []<>[]
[]

Kvartier / N-ære [] []
<>
[] []

Eks.

```
[projekt]* -<>- *[sprog]
          *[person]
```

En person kan arbejde med flere sprog på samme projekt.

```
[Projekt]*-| - - [Sprog]
          *[Person]
```

Ved Associering er Klasserne LIGE:

```
[Låner]-----[Bog]
          udlån
```

6.1.1.4. Aggregation

Ved Aggregering er der HELHED (Flyvning) og DELE (Passagerer og Bagage):

```
[Flyvning]1<>-----*[Passagerer]
          1<>-----*[Bagage]
```

At have en flyvning gør, så vi snakker om Passagerer og Bagage. Flyvning kan godt eksistere uden Passagerer og Bagage. DELE bliver ikke nedlagt hvis HELHEDen bliver det.

Skrives der ikke noget ved HELHEDen regner man med at der 'står' 1.

6.1.1.5. Komposition

Ved Komposition (hård aggregering) er der (Sæder) EJET AF (Fly):

```
[Fly]
  □
  |
  *[Sæder]
```

Sæderne kan ikke eksistere uden Fly, men Fly kan godt eksistere uden Sæder. (var der 1..* i stedet for * ville Fly ikke kunne eksistere uden et sæde). Forsvinder EJER, bliver underliggende nedlagt.

Skrives der ikke noget ved HELHEDen regner man med at der 'står' 1.

6.1.1.6. Abstrakte klasser

- = kan ikke instantieres
- = skrives med kursiv eller understreg eller {abstract}
- = skal have en konkret underklasse
- = anvendes til genbrug – 1. beskrivelse – 2. kode

| |
|------------------------|
| Konto {abstract} |
| kontonummer |
| getEjer |
| udregnrente {abstract} |



6.2. Udviklingsmetoder

6.2.1. Rumbaugh – udviklingstrin

- system ide
- analyse
- systemdesign
- klassesdesign (SK)
- implementering (realisering)
- test
- træning (implementering)
- udrulning (implementering)
- vedligeholdelse

6.2.2. Udviklingslivscyklus

- vandfaldsmodellen: analyse -> design -> realisering
- iterativ udvikling: aktiviteterne gentages for mindre dele af systemet

6.2.2.1. Hvor kommer det nye system fra

- ny funktionalitet
- strømligning (få fjernet nogle irriterende ting)
- simplificering

- automatisering
- integration (ex. af et system i et andet)
- analogier
- globalisering

6.2.3. Hvad er en god systemdefinition

En der kan give svar på:

- hvem laves applikationen for
- hvilke problemer skal app løse
- hvornår vil app blive brugt
- hvornår er der brug for app
- hvorfor er der brug for app (hvorfor kan noget andet ikke bruges)
- hvordan vil app virke

7. SD, 2. SEMESTER

7.1.1.1. Arlow-bog

Website: www.umlAndTheUnifiedProcess.com

7.2. Systemudviklings niveauer

- Værktøj: Software der understøtter udviklingen
- Teknikker: Beskrivelsesmåder der understøtter udvikling af IT-systemer (fx UML)
- Metoder: Sammenhængende fremgangsmåder for anvendelse af teknikkerne (fx finde klasser)
- Modeller (udviklingsprocesser): Overordnede planer for gennemførelse af udviklingsprojekter (fx UP, XP) (skal instantieres, dvs. tilpasses fra projekt til projekt)

Planerne beskriver faser og faselinier (baseline).

7.2.1.1. Aksiomer (grundantagelser)

- Use Case (UC) drevet
 - o Use Casene er det overordnede drivmiddel for udviklingen
 - o Tæt sammenhæng til forretningskrav → brugertilfredshed
 - o Planlægning med blandt andet udgangspunkt i UC
 - o UC kan følges gennem hele udviklingen (UC → sekvensdiagram → skærmbillede → program)
- Risikodrevet
 - o Eliminere de største risici først (gem ikke det sværeste til sidst)
 - o Vurdering af de enkelte risici i et skema
- Arkitekturcentreret
 - o Udviklere har en overordnet forståelse af systemet
 - o Planlægning ud fra arkitektur

- Iterativ: planlægning i miniprojekter
 - o Hurtigere resultater (varieret arbejde)
 - o Let at integrere ændrede krav
 - o Iterations længde 2-6 uger
- Inkrementel: gradvis udbygning af systemet (ikke en prototype)
 - o Hurtig afprøvning af arkitektur
 - o Brugbare resultater

7.2.1.2. Risici i Kajs biler

- skyde over målet
- svært at gøre til flerbrugersystem
- erfaren bruger forsvinder

7.3. Faserne i UP

4 overordnede faser

- Inception (hvad er det kunden vil have lavet)
 - o 1-2 iterationer
 - o Sikring af at det er et fornuftigt projekt (gerne gøre opmærksom på sin tvivl, men ikke dømme eller komme med personlige holdninger)
 - o Hovedvægt på krav (requirements)
- Elaboration (bearbejdning/udbyde/udvide, hvad foregår der virkelig)
 - o 2+ iterationer
 - o Fordybe sig i problemstillinger
 - o Programmering af de første dele
 - o Hovedvægt på krav og analyse
- Construction
 - o # iterationer er afhængigt af systemstørrelse
 - o Programmerer systemet
 - o Hovedvægt på design, programmering og test
- Transition (overførelse/overgang)
 - o 1-2 iterationer
 - o Forberedelse af ibrugtagning
 - o Hovedvægt på test

7.3.1.1. Inception (resultat/mål)

- Interessenterne er enige om projektets hensigt
- Afgrænsning er på plads
 - o Beskrivelse af overordnede funktionelle krav
 - o 10-20 % UC er overordnet beskrevet

- Hvad skal ikke med i systemet
- Risikovurdering foretaget
 - Beslutning om håndtering af kritiske risici
- Skitse til arkitektur
 - Evt. arkitektureksperimenter
- Bekræftelse af projektets tekniske muligheder (fx enighed om en plan B)
- Konsekvensanalyse beskrevet (business case)
 - Beskrivelse af fremtidig forandring
 - Økonomisk vurdering
- Overordnede tidsplan er accepteret

7.3.1.2. Elaboration (resultat/mål)

- En robust/kørende arkitektur
 - En mindre del af systemet fx bygget op med basis i arkitekturen
 - Forståelse af de ikke funktionelle krav
 - Den programmerede del skal være testet
- En UseCase model
 - 80-90% af UC er overordnet beskrevet
 - Detaljeret beskrivelse af en mindre del
- En statisk model
 - Det meste af analyseklassemodellen er beskrevet
 - De fleste analyseklasser er beskrevet
 - En mindre del af modellerne er videreført til designklassemodel
- En dynamisk model
 - Udvalgte UC er analyseret ved anvendelse af sekvensdiagrammer
 - De udvalgte UC er beskrevet i designudgave
- Risikovurderingen er justeret
- Projektplan udarbejdet i passende detalje og accepteret af interessenterne

7.3.1.3. Construction (resultat/mål)

- Et kørende system af den ønskede kvalitet
 - Systemet er testet
 - Systemet er godkendt af brugerne
 - Systemet er klar til implementering
- Dokumentation er klar
 - Dokumentation svarende til krav, analyse, design og programmering er udarbejdet
 - Brugerrettet dokumentation er udarbejdet
- Evaluering af projektet

- Opfølgning i forhold til projektplan

7.3.1.4. Transition (resultat/mål)

- Et implementeret system
 - Evt. i form af en Beta udgave
 - Plan for implementering
- Fuldstændiggjort system
 - Fejl og mangler er udbedret

7.4. Dokumentation

- Anvendelsen af dokumentation (hvad bliver det brugt til)
 - Aftale grundlag
 - Arbejdsredskab
 - Holde styr på viden
 - Styringsredskab
 - Hvor langt vi er nået
 - Vedligeholdelse
- Arbejde med dokumentationen
 - Iterativ udvikling
 - Dokumenterne er i princippet ikke færdige før til sidst
 - "Just enough"
 - Undlad bevidstløs dokumentation, lav det man får noget ud af
 - Diagrammer <-> beskrivelse
 - DIAGRAMMER MÅ IKKE STÅ ALENE
 - Diagrammer skal altid beskrives
 - Krav om konsistens
 - Brug betegnelser ens overalt (kan evt. lave en 'ordbog')
 - Sørg for sporbarhed
- Dokumentationsperspektiver
 - Dokumentation af de enkelte flows
 - Skal hænge sammen til sidst
 - Ufuldstændigt undervejs
 - Dokumentation for faserne
 - På tværs af work-flow dokumentation
 - Dokumentation af processen

7.5. Eksperimenter

Er minimale prototyper, der er til at smide væk når de har tjent deres formål.

Et eksperiment vil enten være vertikalt (dække en enkelt funktion) eller horisontalt (give et overblik, men har intet dybdegående).

Når man skal lave eksperimenter bruger man sjældent reelle programmeringsværktøjer. Grunden er at man så har for vane at gøre 'for meget' ud af det. Et eksperiment skal helst ikke tage mere end mellem et par timer og en dag at lave.

Eksperimenter kan udformes på papir, hvis man ønsker det, eller i et simpelt program (bl.a. PowerPoint er velegnet).

7.6. Projektarbejde

- eksempler
 - o folkeskolen
 - o HTX/HHX store opgaver
 - o MDL (multimediedesigner)
 - o HHÅ (handelshøjskole)
 - o DMU

7.6.1.1. Definition: udforskende læring

- problemformulering
- opstilling af løsningsstrategi og valg
- løsning på opgaverne
- tværfaglighed

7.6.2. Projektarbejde på 2. semester

- % problemformulering
Givet/udleveret problem
- % strategi
Givet/udleveret – brug Arlow
- + løsning af problem
- + tværfagligt

7.6.2.1. Mål

- Indlæring af teori
- Forståelse for sammenhænge mellem fag
- Danne grundlag for eksamen

7.6.2.2. Samarbejde

Formalier

- Ambitionsniveau (hvad man regner med at opnå)
- Arbejdstid (forventninger til hvor meget, hvor og hvornår der skal arbejdes)
- Håndtering af uenighed (fagligt kan man beskrive/forklare hvad man mener)
- Et medlems manglende indsats (hvad gør vi – ringe, kage ...)

7.6.2.3. Lærernes rolle

- sparring (faglige problemer)
- problematisering ('hvad hvis ...?')
- hjælpe med at holde overblikket
- indgå i teori diskussioner
- % deltagere i grupperne

8. SUM, 3. SEMESTER

Dækker over både 3. og 4. semester. Bruges i forbindelse med 4. semesterprojektet. Eksamen i begge semestre efter 4. semester.

8.1.1. Fagets indhold

- Kvalitet (især 3. semester)
- Proces <-> produkt
- Flere trin i livscyklus
 - o (i forhold til SD 1.-2. semester: Krav → Analyse → Design → Programmering → Test)
 - o Forundersøgelse
 - o Implementering (ibrugtagning), vedligehold
- Udvidet analyse og design

8.2. Kvalitet i systemudvikling

- Funktionelt: Hvad man kan gøre med en genstand
 - o Cykel: Transport af en, evt. flere, personer
 - o Termoflaske: Holder væsker varme/kolde (konstant temperatur)
- Kvalitet: Egenskaber ved genstanden
 - o Cykel: Holdbarhed, design, adgang til reservedele, antal gear, specialudstyr, racer (eller ej), billig <-> dyr (pris), let <-> tung (vægt), let at reparere
 - o Termoflaske: Stor isoleringsevne, solid, rumfang, tæt, let at transportere, let (vægt), design
- Kvalitetsmåling
 - o Beskrive kvalitet, så det er målbart (optimalt)
 - o Kvalitet er subjektivt
- Opfyldelse af kvalitet
 - o Angive mål: Beskrive ønskede kvaliteter
 - o Forventninger: Ubeskrevne kvaliteter
- Kvalitetsstyring
 - o Opstilling af kvalitetsmål
 - o Kvalitetssikring: Indbygge ønsket kvalitet undervejs i konstruktionen

- Kvalitetskontrol: Når produktet er færdigt, vurderes om det har de ønskede kvaliteter
 - Test, review

8.2.1. Opgave 1

Kvalitet i produkt

Tænk tilbage på det, I hidtil har arbejdet med i uddannelsen, f.eks. projektet på 2. semester.

Beskriv, hvordan følgende er foregået (eller måske burde have foregået!), og giv konkrete eksempler på indholdet:

- Opstilling af kvalitetsmål for det system, der skal udvikles
- Arbejdet med at opnå disse mål undervejs
- Kontrollen af, om det ønskede kvalitetsniveau er nået

Opstille kvalitetsmål

Snakkede i gruppen om hvordan vi følte om det (primært de 5 punkter om brugervenlighed), efter en diskussion var vi nået frem til hvordan vores prioriteter var ud fra argumenter og tidligere erfaringer.

- Brugervenlighed
 - Prioritering (fx MoSCoW)
 - Analyse el. kravspecifikation (IF)
 - Konkret beskrivelse - målbar

Opnå mål undervejs

Vi har ikke bevidst arbejdet for at opnå målene undervejs. Kun hvis der har været åbenlyse stridigheder i forhold til målene har vi gjort noget specifikt for at opnå/nærme os dem.

- Efterligning (af nuværende system)
- Regler for design (gestaltlove)
- Tale med brugeren (kend brugeren)
- Ensartethed

Kontrol

Ingen af os har reelt testet på kvalitetsmålene. Dette har dog været grundet manglende testgruppe og tidsnød. Tests til kvalitetskontrol har dog været udarbejdet.

De tests der har været udført har haft til formål at afprøve funktionaliteten.

- Test
 - Fx efter den konkrete beskrivelse (checke mod ønsket mål)

8.3. Produkt og proces i systemudviklingen

- Definitioner
 - Produkt: Det endelige system med tilhørende dokumentation samt mellemprodukter
 - Mellemprodukt: prototype i papir, kravspecifikation, analyserapport, klassediagram, use-case specifikationer

- Proces: Det samlede arbejde med at nå frem til produktet
- Produkt rettet
 - Analyse: Undersøgelse og systematisering af bestående forhold
 - Fx kravspecifikation, analyseklassediagram, use-case model, sekvensdiagram
 - Design: Beskrivelse af visioner om en fremtidig situation
 - Fx use-case, tilstandsdiagram, kravspecifikation, designklassediagram, skærmbillede forslag
 - Realisering: Virkeliggørelsen af visionerne
 - Fx program, brugervejledninger, manual
 - Ved implementering, fx uddannelse/oplæring
- Proces rettet
 - Vurdering: Se på og reflekter over forløb (fx ved tidligere projekter)
 - Fx timeopgørelse, statusrapport
 - Planlægning: Beskrivelse af ønsket/muligt forløb af projektet
 - Fx projektplan, tidsbeskrivelse, estimering
 - Regulering: Regulering af forløbet undervejs
 - Fx nye arbejdsmetoder, forhandlinger, kurser i nye værktøjer/metoder/sprog

8.3.1. Opgave 2

Kvalitet som et relativt begreb

Prøv at finde eksempler på, at den samme kvalitet ikke er nødvendig i to forskellige systemer

Brugervenlighed vs. Let at lære

Backup

Vedligeholdelsesvenligt (lagdeling/arkitektur) vs. Hurtigt

8.4. Kravspecifikation

8.4.1. Hvad er det?

- Samling af krav til opgaver i det kommende system
- Liste over kundens behov (som skal dækkes af det nye system)
- Krav prioriteres (MoSCoW¹) og nummereres
- Giver kunde og udvikler fælles opfattelse/forståelse af opgaven → facitliste ved accepttest
- Udformning
 - "Med systemet skal brugeren kunne:"

¹ Must have, Should have, Could have, Would like have

- "1) Oprette og redigere kundeoplysninger"
 - "1.a) Systemet skal kunne fjerne inaktive kunder"

8.4.2. Definition

Aftale om projektets resultat mellem udvikler og rekvirent/kunde

- Krav skal kunne styres (det kan være at et krav ikke kan overholdes indenfor aftalte tid/pris) da
 - Omkostningerne kan blive for høje, krav fjernes (W og C)
 - Krav kan formuleres bedre
 - Krav kan stride mod hinanden
 - Der kan være ændringer i kundens behov
- Overordnet indhold: Beskrivelse af de krav, det kommende system skal opfylde
- Krav specifikationen skal kunne revideres disciplineret
- Parterne skal kunne måle resultatet (systemet) ud fra kravspecifikationen

8.4.3. Indhold

- Funktionelle krav: Hvad systemet skal kunne
- Ikke funktionelle krav/kvalitetskrav: Hvordan systemet overholder de funktionelle krav
- Øvrige afleveringer (i forbindelse med kravspecifikationen): Dokumentation (bruger rettet, udviklingsmæssigt)(brugervejledning, installationsvejledning, hotline nummer m.v.), installation, konvertering af gamle data, oplæring af personel, (hvorvidt kildekode skal udleveres samt hvornår)
- Ledelsesrettede krav: Deadline, ophavsret, ansvar, kontrakt
- Hjælp til læseren: Definitioner (fx ordliste), eksempler, diagrammer

8.4.4. Ikke-funktionelle krav

- Daglig brug
 - Sikkerhed (adgangssikkerhed, datasikkerhed)
 - Korrekthed
 - Pålidelighed (opetid)
 - Brugbarhed (brugervenlighed)
 - Effektivitet (behandlingstid, RAM forbrug, disk-forbrug)
- Krav til tilpasning
 - Vedligehold (rettelser i system)
 - Testbarhed
 - Udvidelighed (tilføje nye krav/faciliteter)
- Overflytning
 - Flytte til ny maskine/OS el. lig.
 - Samarbejde med andre systemer (måske flyttes op til dagligt brug)
 - Genbrugelighed

8.4.5. Kvalitet af kravspecifikation

- Korrekt: Ægte krav/behov
 - o Problem: Krav, vi tror, skal være med/brugeres vildfarelse
- Komplet: Alle krav skal være med
 - o Normalt undlader man de mest trivielle
 - o Hellere lidt for detaljeret end det modsatte
- Utvetydig: Må ikke kunne misforstås
 - o Bruger <> bruger / bruger <> udvikler
 - o Ordliste
- Konsistens: Kravene skal hænge sammen indbyrdes
 - o Lange kravspecifikationer kan betyde inkonsistens
 - o Opdeling
- Prioritering efter vigtighed
- Prioritering efter stabilitet
 - o Risiko for ændringer undervejs?
- Modificerbar
- Verificerbar: Test ud fra kravspecifikation
 - o Økonomisk forsvarligt
 - o Ikke-funktionelle krav sværest
- Sporbarhed
 - o Tilbage til forretningsmål (gælder primært for funktionelle krav)
 - o Fremad til analyse, design og kode
- Forståelighed for både bruger og udvikler

8.4.6. Kvalitetsindbygning i kravspecifikationer

- Samarbejde med brugere
- Forståelse for forretningsmål
- Begrebsordbog/ordliste
- Nummerering, prioritering (MoSCoW)
- Iterativ udvikling – løbende uddybning og tilpasning af kravene

8.4.7. Check af kravspecifikationens kvalitet

- Review
- (Test af prototyper / eksperiment)

8.5. Planlægning og styring af test

8.5.1. Planlægningsniveauer

- Strategisk: Ledelsens overordnede principper for test i virksomheden

- Økonomi, image, konkurrenter (hvad der betyder noget for valget)
- Ingen beslutning er også en beslutning!
- Taktisk: Beslutninger i det enkelte projekt
 - Udarbejdelse af testspecifikation
 - Tæt sammenhæng til projektplanen i øvrigt
- Operationelt: De enkelte tests (kvalitetsmåling)
 - Udføres af alle + eksterne

8.5.2. Testspecifikation

1) Teststrategi

- Traditionel: Programmering → Test
- Moderat (V-model): Testplan → Programmering → Testudførelse (anbefalet)
- Radikal: Testplan + udførelse → Programmér (tvivlsom)

2) Systemegenskaber (~ ikke funktionelle krav)

- Hvordan testes disse krav?

3) Ressourcer: Ønskede/mulige testere

- Hvornår skal de bruges?

4) Niveauer: Som fra V-modellen

- Hvilke vil vi have med i modellen

5) Testområder: Identificere det "svære" i systemet

- Behov for ekstra test
- Husk: brugerrettet + teknisk

6) Testenhed: Én eller få vælges afhængig af niveauet

- Fx krav, use-case, skærbilleder, klasse, tilstande

7) Statisk/maskinel: Teoretisk/praktisk

- Om der ønskes at lave en statisk og dermed teoretisk test eller en maskinel og dermed praktisk test

8) Testteknikker: BlackBox, WhiteBox

- Typisk BB over det hele, måske suppleret med WB nogle steder

9) Testdækning: % af valgt testenhed, evt. løbere angivelse ("ca. halvdelen", "alle ikke trivielle")

10) Risici: Tænk udviklingen og brug

11) Uddannelse: Behov for det?

- Evt. brugere der skal være med til test skal muligvis vendes til ideen om hvordan det foregår

12) Planlægning/styring af aktiviteter

- Forretningsgang i test

13) Organisation: Hvem gør hvad i test?

- Testleder udpeges

14) Teknisk testmiljø: Beskrives

15) Afslutningskriterier: Hvornår er vi færdige/tilfredse?

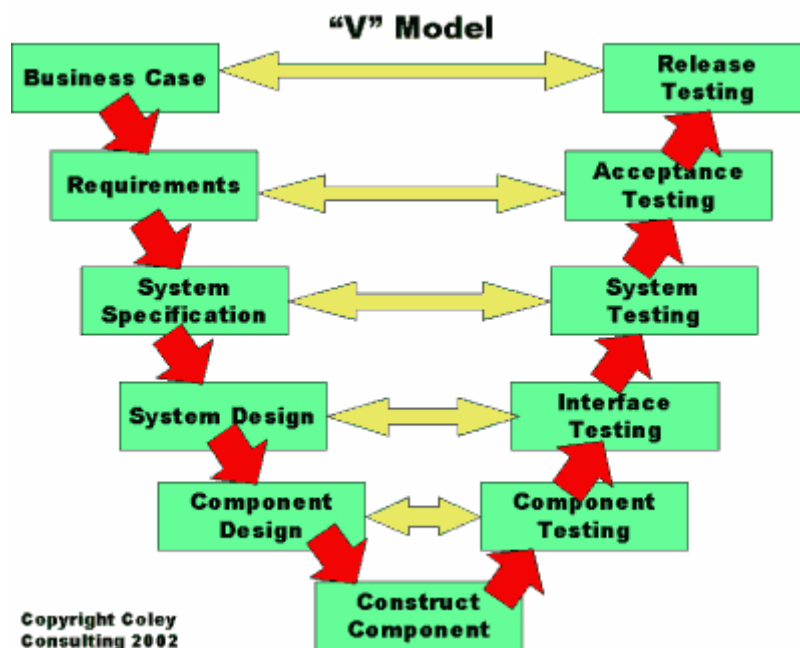
- Fx "når der ikke er flere fejl", "når der ikke er mere tid", "når vi løber tør for ressourcer"

16) Testplan: Grovplan udarbejdes

17) Planlægning af testseminar

- Fx i forhold til medarbejderne, der nu skal udføre en anden funktion (programmering → test)

8.6. V-modellen



Accepttest: noget brugeren laver for at sikre sig at han får det han ville have. Dvs. test på brugerens præmisser

Modultest: tester et modul uden der er forbindelse til et andet (ikke så vigtig som integrationstest)

Integrationstest:

WhiteBox: at koden gør det den skal

8.6.1. Opgave 3 fra dag 2

Testspecifikation

Forestil dig, at du indgår i en gruppe af 4. semester-studerende, der skal gennemføre projektet med at udvikle ovenstående system til grundejerforeningen. I den forbindelse skal I have udarbejdet en testspecifikation med udgangspunkt i skabelonen fra [Vin], afsnit 7.3.

Forestil dig desuden, at gruppen har udnævnt én af jer til projektleder og kontakt til foreningen (formanden), mens resten af gruppen arbejder generelt som systemudviklere. Du kan også forestille dig, at I bruger UP og udvikler i Java, og at jeres arbejdsmetoder og -værktøjer i øvrigt svarer til dem, I kender fra uddannelsen, hvis det har nogen betydning.

Det tilstrækkeligt at skitsere indholdet i de fleste af punkterne i skabelonen. Desuden er du igen velkommen til at gøre dig forskellige antagelser undervejs, hvis det er nødvendigt.

- Strategi: Moderat
- Systemegenskaber: Adgang – check at det er der
- Ressourcer: 2-3 brugere, ønskes: 1 tester/udvikler
 - Ca. 2 dage pr. bruger
- Niveauer: Acceptest, integrationstest
- Testområde: Uklarheder om "beslutninger"
- Testenheder:
 - krav/use-case (accept)
 - sekvensdiagram (integrationstest)
 - skærmbillede (brugergrænseflade)
- Maskinel
- Testtekniker: Blackbox
- Testdækning: 100%
- Risici: Ingen alvorlige
- Uddannelse: Brugere har behov for lidt intro til test
- Planlægning/styring: Følg IEEE-standard
- Organisering: Tester som testleder eller lade projektlederen varetage denne rolle
- Testmiljø = driftsmiljø (test det på brugernes computere)
- Afslutning: ved 0 fejl
- *hop*
- Testseminar undlades grundet projektets størrelse

8.7. Forundersøgelse (FU)

8.7.1. Definition

- Projekt (har afgrænset tid og emne/mål)
 - Uafhængigt af udviklingsprojekt
- Mål afklarende: Hvad vil virksomheden egentlig opnå?
- Problemformulerende: Vigtigt at gøre det klart hvad der starter projektet
- Løsningsangivende: Løsning på problemet (gerne mere end én mulighed)
- Virksomhed (selvvalgt begrænsning)
- Sigte på IT-anvendelser: Andre muligheder, men dér har IT-folk ikke meget at gøre
- Problemstilling som udgangspunkt
- Opgave om forundersøgelse
 - Forundersøgelsens udgangspunkt (nr. 1)
 - Ifølge bogen kan en forundersøgelse indledes, fordi
 - * ledelsen vil forfølge nogle forretningsmæssige mål,

* én eller flere grupper i virksomheden har oplevet et problem,

* nogen har en mere eller mindre konkret ide om, at et problem eller nye opgaver kan klares vha. nye IT-anvendelser

Giv konkrete eksempler på hvert af disse udgangspunkter. Du kan selv vælge en virksomhed – f.eks. Jyske Bank (tænk på 2. semester casen), Erhvervsakademiet (tænk på et Blackboard-agtigt system), en videoudlejningskæde (som Blockbuster), et folkebibliotek eller noget helt andet.

- Ledelsens initiativ: Strategisk beslutning
 - Ledelsen ønsker backup af alle data
 - Ønske om at være først på markedet
 - Blockbuster skift til DVD (fra VHS) + salg
 - STY i Jyske Bank: ønske om at minimere tab
- Mellemlidelses initiativ: Konkrete problemer i arbejdsprocessen
 - Jyske Bank: Problem med beregning (kompleks formel)
 - Papirer der sendes rundt (bøvlede forretningssange)
 - Blockbuster havde problemer med at anbefale film
- Alles initiativ: Ide om ny IT-anvendelse (ofte i form af inspiration fra andre, fx konkurrenter, forskning m.v.)
 - Nettos prisscannere, Kvickly, Bilka
 - Supermarkeds selvbetjening (kassedameløse)
- Forundersøgelsens resultat (nr. 2)

Bogen siger, at resultatet af en forundersøgelse er en rapport, evt. suppleret med prototyper. Tilsammen skitserer de en eller flere visioner om forandring i forhold til teknologi, arbejdsorganisation og de nødvendige kvalifikationer hos de ansatte.

Hvad kan ideen være i at skitsere mere end én vision?

Hvad gør man, hvis der er flere end én visioner – for man kan og vil jo nok ikke realisere alle visionerne!

- Hvorfor mere end én vision
 - Forskellige grupper har forskellige mål (modstridende) hvor løsningsforslagene dog vil overlappe hinanden
 - Større og mere komplet vision samt mere skrabet vision (det nødvendige)
- Hvordan vælges der mellem visionerne
 - Hvad Ledelsen føler bedst
- Prototyper som en del af resultatet (nr. 3)

Hvad er ideen med at lave prototyper som supplement til en rapport under forundersøgelsen? Hvem kan have glæde af den? Hvilken type prototype skal det i givet fald være?

- Hvad er ideen
 - Det kan være at afsender og modtager har for forskellige terminologier/ forskellige måder at se ting på
- Hvem har glæde af den
 - Modtager kan 'se' det han har læst beskrivelsen af → bedre forståelse (og færre misforståelser)
- Hvilken type

| | | |
|--|----|--|
| Smid væk (ofte) idet der kun er tale om en forundersøgelse | <> | til Videreudvikling (knap så ofte) kan være smart fx hvis flere visioner ligger op af hinanden idet den så i det mindste kan genbruges |
| Vertikal (knap så ofte) | <> | Horisontal (ofte) det ofte kan være svært at overskue dybden på så tidligt et stadie |
| Afprøvende (knap så ofte) det er ikke ofte at der er en ide til en konkret opbygning på et så tidligt stadie | <> | Udforskende (ofte) kan også bruges til at få ideer |
| IT baseret | <> | Papir baseret (valg afhænger af tid og målgruppe) |

8.7.2. Resultat af forundersøgelse

- Rapport
 - Skitse af visioner om forandringer
 - Vurdering af konsekvenser (fx cost/benefit analyse)
 - Økonomisk overslag (investerings kalkyle)
 - Strategi/plan for realisering (fx købe system/egen udvikling, tidsrammer)
- Skitse af vision
 - Kort beskrivelse af overordnede krav
 - Opstilling af rammer – hvad er med og hvad er udeladt?
 - Evt. prototype

8.7.3. Viderearbejde

- Valgproces ved mere en én vision
 - Kunden vælger suverænt (ledelse, evt. brugere)
 - Udviklere supplerer med information efter behov
- Udbudsforretning
 - Licitation
 - Indbudte leverandører
 - Egen udvikling
- Realiseringsprojekt (udviklingsprojekt)

- Evaluering (Fik vi det, vi forventede? Var det godt eller skidt?)

8.8. MUST-metoden

MUST = metoder til forundersøgelse i systemudvikling og teorier herom

- Forberedelse
 - Projekt etablering af forundersøgelsesgruppen
 - Socialisering (kan være en fordel at komme væk fra vante omgivelser)
- Fokusering
 - Forstå virksomhedens strategier og udvælge arbejdsområde (produktion/kunde pleje/økonomi ...)
- Fordybelse
 - Forstå arbejdsområdet inklusiv problemer (også dem der har startet projektet)
 - (er det et reelt problem eller et det et symptom?)
- Fornyelse
 - Udvikle visioner + rapportering

8.8.1. Principper i MUST

1 En samlet vision

- (består af 3 ligeværdige dele, dog behøver tidsforbruget ikke være lige fordelt)
- IT-udvikling
 - Funktionalitet, grænseflade, sammenhæng til andre systemer
- Organisatorisk udvikling
 - Forsalg til fremtidig organisering af arbejdet
- Kvalifikationsudvikling
 - Den enkelte medarbejders mulighed for at klare arbejdet fremover

2 Reel brugerdeltagelse

- Aktiv brugerdeltagelse/involvering i forundersøgelsen
- Gensidig læreproces
 - IT-konsulenter lærer om virksomheden
 - Medarbejdere i virksomheden lærer om IT
- Moralsk/politisk
 - Brugere har indflydelse (generelt) på deres arbejde (i hvert fald i Skandinavien)
- Resultatet af FU har større chance for accept

3 Arbejdspraksis skal opleves

- Beskrevet arbejdspraksis
- Fortalt arbejdspraksis
- Reel/oplevet arbejdspraksis
- Alternativt: læse materiale, interviews m.v.

- Imødegår dilemmaet mellem holdning og adfærd
- Understøtter ideen om gensidig læring
- Retningslinjer
 - 1. håndserfaring
 - Resultat: Relevante, forståelige beskrivelser af arbejdsgangen
 - Beskrivelse, ikke 'foreskrivelse'
 - Fra observandens perspektiv

4 Forankring i virksomheden

- Sikre realismen i projektarbejdet
 - Give information, sikre opbakning og forståelse hos brugeren
- Forankring hos
 - Styregruppen (ledelsen), brugere/ansatte, IT-ansatte
- Metode
 - Møder, skriftlig information

8.8.1.1. Opgave 5 fra sidst

Problemer ved at undlade forundersøgelse

Alternativet til at lave en forundersøgelse er at kaste sig direkte ud i et realiseringsprojekt.

Giv mindst 3 bud på, hvad der kan gå galt i realiseringsprojektet, hvis man har undladt forundersøgelsen.

- Forkert system udvikles
- Fejlestimeringer, fejl skøn økonomisk
- Forkert bemanning
- Ukorrekte krav til systemet
 - Fx sikkerhed
- Opbakning – modstand mod forandring

8.8.2. Forberedelse

Formål

- Afklare udgangspunktet
- Skabe forpligtelser for resten af FU

Udgangspunkt

- Mere eller mindre præcist formuleret kommissorium (rammer for hvad der skal laves)

Aktiviteter

- Afklaring af rammer, indhold
- Organisering af FU
- Planlægning af FU
- Socialisering

Resultat

- Projektgrundlag: præcisering af indholdet af kommissoriet
- Plan for forundersøgelsen (hvem, hvad, hvornår, ...)

8.8.3. Fokusering: Strategi analyse

Udgangspunkt

- Projektgrundlag og projektplan

Formål

- Afklare/afstemme formålet for projektet med virksomhedens strategier/mål

Aktiviteter

- Strategier findes, evt. udvikles
- Udvalgte arbejdsområder

Resultat

- Strategianalyserapport

8.8.4. Fordybelse: Analyse af arbejdsgange

Formål

- Finde og prioriterer mål, problemer og behov i arbejdsgangene

Udgangspunkt

- Strategirapport (specielt valg af arbejdsområde)

Aktiviteter

- Dataindsamling (interview, observationer)
- Analyse/refleksion
- Præsentation (mundtligt/skriftligt/rige billeder)

Resultat

- Analyserapport (fokus på problemer)
- Evt. beskrivelser af arbejdsgange

8.8.5. Fornyelse

Formål

- Forberede brugere og ledelse på forandringer og skabe et realistisk beslutningsgrundlag

Udgangspunkt

- Analyserapport (behov og problemer)
- (beskrivelser af arbejdspraksis)

Aktiviteter

- Vedrørende IT-systemer
- Forandringer i organisationen og kvalifikationer
- Konsekvensanalyser/konsekvensberegninger (fx cost-benefit)
- Strategi for realiseringsprojektet

Resultat

- **Udvikle visioner + rapportering**

8.8.6. Opgaveløsning med brug af MUST

En undervisningsinstitution, der udbyder videregående uddannelser og efteruddannelse på tilsvarende niveau, ønsker at effektivisere den information, som studerende og medarbejdere har brug for at udveksle i forbindelse med studierne. Mere præcist drejer det sig om skemaer, undervisningsplaner, lektionssedler, opgaver, opgavebesvarelser, praktiske oplysninger, eksamenstilmeldinger og -resultater, samt interne nyheder af enhver art.

Opgaven går nu ud på at finde frem til forslag til, hvordan dette kan løses. I forventes naturligvis at anvende MUST-metoden fra [Bød]. Hvis I, i forbindelse med løsningen af denne opgave, har brug for mere præcise oplysninger, er I velkomne til at bruge Erhvervsakademiet som eksempel.

MUST-metoden omfatter 4 faser, som hver lægger op til en opgave.

1. Forberedelsesfasen

Fasen omfatter blandt andet følgende aktiviteter. For hver aktivitet er givet nogle spørgsmål, som I bedes tage stilling til.

- **Kortlæg interessenter:** Hvem vil være interesseret i dette forundersøgelsesarbejde? For mindst tre interessenter skal I give et bud på, hvordan de kan påvirke projektarbejdet, og hvordan de selv bliver påvirket af det. Overvej også, hvad succeskriterierne kunne være for interessenterne (dvs. hvad der skal til, for at de vil opfatte projektet som en succes).
- **Udvælg deltagerne:** Hvem vil det være rimeligt at inddrage i denne forundersøgelse? Tænk især på brugerrepræsentanter. Hvor mange vil I mene, der skal være med i en effektiv gruppe?
- **Etablér projektgruppen socialt:** Giv forslag til (realistiske!) måder at ryste gruppen sammen på!
- **Start forankring:** Hvad vil I foreslå, der kan gøres her i starten af projektet for at sikre, at forundersøgelsesarbejdet samlet set vil blive accepteret?

2. Fokuseringsfasen

Overvej omfanget af fokuseringsfasen ud fra figur 5.1. Hvilken situation vil I ende i sandsynligvis?

Prøv at kigge lidt i skolens strategi for it og pædagogik (kan findes sammen med dagens lektionsseddel på Blackboard). Kan I finde belæg for, at udgangspunktet for denne forundersøgelse er fornuftigt?

(Overvej, om du har behov for at udskrive strategien, der fylder 17 sider!)

3. Fordybelsesfasen

Overvej også her omfanget af fasen, denne gang med udgangspunkt i figur 6.1. Hvilken situation vil i sandsynligvis ende i?

Hvordan vil I finde frem til problemer i den nuværende arbejdspraksis? Husk på princippet om, at arbejdspraksis skal opleves! Giv forslag til et par konkrete aktiviteter.

Kan I selv – ud fra jeres egne oplevelser – se problemer, som det vil være vigtigt at inddrage i den kommende forandringsproces?

4. Fornyelsesfasen

Kender I nogle teknikker til at få nye ideer med? Beskriv dem kort.

Har I på nuværende tidspunkt ideer, der kan bidrage til at forbedre arbejdsgangene og dermed indgå i en vision? Husk, at der er tale om en samlet vision med elementer af både IT-løsning, organisatorisk tilpasning og forøgelse af den enkeltes kvalifikationer.

8.8.6.1. Opg. 1, Forberedelsesfasen

- Interessenter (påvirkes af / påvirker)
 - Udviklere/support
 - Undervisningsministeriet (0 / 0-1 (4))
 - Brancheorganisationer
 - Administration/Ledelse (1 / 5)
 - Kan påvirke overblik set inde bag fra systemet
 - Ønsker at det er let at bringe oplysninger ud til lærere og studerende. Være billigt, fungerer, fleksibelt og ingen der klager
 - Lærerne
 - Påvirke ting omkring filer (upload og valg af synlighed)
 - Ønsker at systemet skal være let at håndtere, det skal være let at ligge ting ud til eleverne samt lave ændringer (f.eks. i skemaer)
 - Elever/Studerende (5 / 1-2)
 - Kan påvirke hvordan systemet 'logisk' hænger sammen og hvor let det er at navigere
 - Ønsker at systemet skal være let at få oplysninger fra (overskueligt). Være opdateret, hurtigt og let at finde rundt i.
- Deltagere
 - To fra hver af læregruppen og administration.
 - En elev fra hver uddannelsesgruppe
- Socialt
 - Bruge en eftermiddag på at spise sammen (evt. kombineret med at bowle)
 - Projektgruppe-weekend i hytte (socialt + fagligt)
- Forankring
 - Melde ud til hver deres bagland hvad der sker/kommer til at ske → mail/personligt fremmøde
 - Bagland skal vide at de kan gå til repræsentanter/projektgruppe hvis de har en holdning
 - Lærere tilsvarende
 - Materiale på fællesområde (BB, AgerNet)

8.8.6.2. Opg. 2 Fokuseringsfasen

- Ud fra figur 5.1 vil udviklingen have i situation 1 eller 3. Sandsynligvis situation 1 idet der er tale om et informationsdelingssystem

8.8.6.3. Opg. 3 Fordybelsesfasen

- Opleve arbejdspraksis
 - Observere lærere og elever bruge det nuværende system til deres daglige opgaver
 - Stille opgaver som de bør kunne finde/gøre ('kunstige opgaver')
 - Interviews
- Problemer
 - Fraværd og informations fra undervisningen (→ Pod cast af undervisning)
 - Finde information (fx referat fra KU møder)
 - Uoverskueligt system

8.8.6.4. Opg. 4 Indvendinger (imod ideer)

- Svært at vende sig til noget nyt
- Tager for lang tid
- Det er for dyrt
- Det er ikke relevant
- Ved hvad vi har men ikke hvad vi får
- Det er ikke muligt at gennemføre

8.9. Brugerinddragelse i systemudvikling

8.9.1. Problemer med opbakning (generelt)

8.9.1.1. Indvendinger (imod ideer)

- | | |
|--|--|
| <ul style="list-style-type: none">- Vores problem er anderledes- Vi har allerede prøvet det- Vi har ikke tid til det- Der er ikke nogen, der kan hjælpe os- Vores system/organisation/gruppe er for lille til det- Vi plejer at gøre det på den måde: ...- Vores nuværende metode har vist sig at virke gennem årene – den kan man stole på- Det er upraktisk- Ideen er forud for sin tid- Ideen hører til i fortiden- Vi er ikke klar til det endnu- Vi har prøvet for meget af den slags i den seneste tid- Man kan ikke lære gamle hunde nye tricks | <ul style="list-style-type: none">- Vores unge, progressive afdeling har ikke brug for det- Det vil kræve store investeringer- Det vil aldrig kunne betale sig- Hvis det ikke koster noget, hvordan forestiller du dig så, at det kan bruges til noget- Det er for drastisk- Det er næsten ligesådan som vi gør nu- Det ser godt ud på papiret, men det vil ikke virke i praksis- Det overholder ikke regler/standarder/politikker- Bestyrelsen/ledelsen vil ikke bryde sig om det- Vores nuværende løsning virker, hvorfor så prøve noget nyt? |
|--|--|

8.9.1.2. Mulige modargumenter

- Aftale at man ikke må kritisere ideerne første gang de kommer op
- Brainstorming
- Succeshistorier
- Konkretisering af ideerne
- Større fra de indflydelsesrige (politik)
- Positiv brainstorm (der må ikke komme negative ting under brainstormen og alle ting skal tages seriøst)
- Skræmmekampagne for det nuværende system (hvad der sker hvis vi ikke fornyer os)
- Fremtidsværksted
- Ekstern inspiration
 - Messer
 - Virksomhedsbesøg
 - Surfe Internettet
- Metafor (for at få sin egen situation på afstand)
 - Hvad nu hvis undervisning var fabriksproduktion?
- Eksperimenter med prototyper

8.9.2. Modstand mod forandring

Reaktioner

- Aggression: Forsøg på sabotage
- Projektion: Bruger skyder skyld for alle problemer på systemet
- Undgåelse: Brugere forsøger at løse opgaverne uden brug af systemet

Årsager (brugernes opfattelse)

- IT-afdelingen har for stor magt
- IT-folk har for stor status/løn/for mange rettigheder

Årsager (reelle(?))

- For dårlig kommunikation
- For dårlige brugergrænseflader
- Ingen/dårlig oplæring
- Manglende ledelsesopbakning
- Manglende inddragelse af brugere

8.9.3. Brugerdeltagelses niveauer

Konsultation: Passiv deltagelse

- Udviklere tager initiativ
- Ofte ret tidligt i forløbet

Repræsentativ deltagelse

- Et lille antal brugere deltager mere aktivt i udviklingsprocessen

Fuld deltagelse

- Alle kommende brugere deltager aktivt (normalt kun hvis gruppen er lille)

Brugerne som udviklere (ingen IT-udvikling)

- Kræver rimeligt IT-kendskab
- Har tid
- Skal kunne betale sig
- Kræver ret simple systemer

8.9.4. Områder med mulighed for aktiv deltagelse

Forundersøgelse: MUST kræver det

- Repræsentativ, evt. fuld deltagelse

Kravsøgning/kravbeskrivelse

- Alle niveauer er mulige, men repræsentativt/fuld deltagelse er mest hensigtsmæssig

Analyse

- Dele heraf kan brugerne med fordel deltage aktivt i (fx use-case specifikation)

Design

- Aktiv deltagelse ved brugergrænsefladedesign, ellers ingen deltagelse

Programmering

- Kun ved simple udviklingsværktøjer, ellers ikke

Test

- Repræsentativ/fuld deltagelse ved accepttest (både udvikling og gennemførelse af)

Implementering

- Både og!
- Små systemer: Brugere klarer det selv
- Større sammenhæng: Udviklere/IT-afdeling

Vedligehold

- Brugere kan deltage i prioriteringen af ændringsønsker

Projektstyring

- Brugerrepræsentanter i styregruppen

8.10. Kvalitetsindbygning

8.10.1. Måder

- Brug af metode, værktøjer, teknikker
 - Fx UML, prototyping
- Udnytter erfaringer

8.10.2. Mønstre

- Design mønstre
- Kendskab til mønstre
- Udvælgelse og tilpasning til konkret problemstilling
- Historisk
 - Arkitektur
- Andre områder med mønstre
 - Analysemønstre
 - Interaktionsmønstre
 - Arkitekturmønstre
 - Samarbejde (f.eks. projektgruppe)

8.10.3. Analysemønstre

- Klassediagram
 - Rollemønstre
 - Samling (composite)
- Use-cases

8.10.4. Mønster-mønster (!)

- Navn
 - Nødvendigt for kommunikation
- Problem
 - Bør kunne bruges i mange situationer, gerne suppleret med eksempler
- Løsning
 - Skitse-mæssig form
- Fordele/ulemper

8.10.5. Brugergænsefladedesign

8.10.6. Molic (repetition)

- Kvalitetskrav: De fem krav til brugervenlighed
 - Check af brugervenlighed
- Lidt tynd på 'en metode'
 - Mange gode råd

8.10.7. Metode (Benyon, Turner & Turner)

- Scenario-baseret metode
- Mellemprodukter
 - Brugerfortællinger: anekdoter, forløb, tanker, observationer
 - Systemuafhængig

- Begrebsmæssige scenarier: Sammenskrivning af en gruppe brugerfortællinger i abstrakt form (lidt som use-case i UP)
 - Systemuafhængig
- Konkrete scenarier: Beskrivelse af interaktion med systemet ud fra strukturen i begrebsmæssige scenarier (inkl. overordnede designbeslutninger)
 - Systemnære
- Use-case: Systemnære use-cases
 - Systemnære
- Kravspecifikation (den overordnede)
- Scenariosamling: Samling af alle de begrebsmæssige scenarier – beskriver det samlede system (a la Kajs biler)
- Begrebsmæssig model: Metafor for brugergrænsefladen og interaktionen (fx skrivebord)
- Designprog: Håndbog for stilen af elementer i interaktionen
- Processer/aktiviteter
 - Abstraktion: Fjernelse af de ubetydelige/for specifikke elementer, samling af flere fortællinger
 - Specifikation af designbindinger (krav/valg)
 - Formalisering af design: Omformning til use-case
 - Kravopsamling: Forstå, beskrive krav og ideer
 - Prototyping, belysning af ideer, evaluering
 - Eksperiment for at finde designprog og medvirke til fysisk design
 - Begrebsmæssigt design: Beskrivelse af de generelle brugergrænsefladeobjekter og aktioner (handling)
 - Fysisk design: Beskrivelse af den endelige grænseflade

8.10.8. Begrebsmæssigt design

1) Analyse af hvert begrebsmæssigt scenario:

a. Finde specifikke aktioner og højniveau aktivitet

| Aktivitet | Delaktivitet | Aktion | Objekt | Evt. kommentar |
|------------------|-----------------------|---------------|---------------|----------------|
| Søg efter nummer | Gå til søgefunktion | Gå til | Søgevindue | |
| | Indtast søgekriterier | Taste [tekst] | Søgekriterier | |
| | | Sæt i gang | Søgeresultat | |
| | | | | |

2) Udtræk objekter/aktioner fra scenarier

- a. Brugergrensefladeobjekter + synlige domæneobjekter
- b. Eksempel på objekter: søgevindue, bog, bogliste, printindstillingsvindue, tekst, billede, kamp

- c. Eksempel på aktioner: gå til, indtast, vælg, godkend, start, afbryd
- 3) Samle klasser og aktioner på tværs af scenarier
 - a. Tæl antal forekomster
- 4) Ryd ud
 - a. Kombiner de elementer (aktion el. klasse), der logisk set er ens

8.10.9. Mønstre i interaktion (eksempler)

Løser nogle problemer, på en bestemt måde.

- Drag and drop
- Menu med højreklik
- Scrolling
- Drop down menu
- Grå til inaktive menupunkter
- Klappe menuer sammen
- Pop-up vinduer

8.11. Fysisk design (del af interaktionsdesign)

- En aktivitet i interaktionsdesign
- Indhold
 - Strukturering af interaktion i logiske forløb
 - Tydeliggørelse af muligheder
 - Overførsel af det begrebsmæssige design til konkrete skærbilleder og anvendelser
- Komponenter
 - Operationelt design: Hvordan gør man, indholdets struktur
 - Repræsentationsdesign: Udseende, layout
 - Interaktionsdesign: Fordeling, menneske <=> maskine
- Mønstre i interaktion
 - Genbrug er godt
- Beskrivelse af interaktionsforløb
 - Tilstandsdiagram (godt ved mødes)
 - Navigationsdiagram
- Designsprog
 - Samling af standardelementer, der bruges mest muligt
 - Farvevalg, input måde (mus/tastatur)
 - Principper for anvendelser
 - Fx skabeloner
 - Sammenhæng mellem brugssituation og reglerne ovenfor
 - Alt dette kan dokumenteres i en interaktionsdesignhåndbog

8.12. Mønstre i analyse

- Use-cases og mønstre
 - Samling af use-cases
 - Den enkelte use-case (udefra)
 - Den enkelte use-case (internt → specifikation)
 - Use-case sammenhænge (include og extend)

8.12.1. Use-case specifikation

- Målbare startbetingelser:
 - Sørg for at use-casen altid har et klart udgangspunkt = hændelse
- Trin Med Homogent Og Passende Niveau (level steps)
 - For detaljeret eller overordnet er ubrugeligt, et miks mellem de to kan også mindske forståelsen
- Tydelig Aktør Målopfyldelse
 - Resultatet af use-case skal være klart og skrevet til brugeren (gælder også de enkelte trin)
- Fremadskridende Trin
 - Undgå beskrivelse i specifikationen, der ikke bruger aktøren videre
- Teknologi Neutral
 - For meget (beskrevet) teknologi binder udviklerne unødigt og gør så det ikke kan realiseres på flere måder

8.12.2. Use-case sammenhænge

- Delt Beskrivelses Sekvens (common subbehavior)
 - Flere use-cases indeholder samme ensartede sekvens
 - De oprindelige <<include>> → ny use-case
- Afbrydelser Som Extensions
 - En use-case har flere steder i specifikationen variationer, der alle skyldes et bestemt forhold (det er dette forhold, der betragtes som afbrydelse)
 - Ny use-case med specialsituation og bruger <<extends>> → oprindelige
- Forfremmet Alternativ
 - En del af en use-case er ret kompliceret
 - Placér det komplicerede i en selvstændig use-case med <<include>>

8.13. Design af (logisk) arkitektur

8.13.1. Logisk arkitektur

- Overordnet organisering af softwareklasserne
- Strukturering ved hjælp af lag, packages, delsystem

8.13.2. Arkitekturanalyse

- Identifikation og analyse af ikke-funktionelle krav (med betydning for arkitekturen)
 - Fx performance, sikkerhed (funktionsagtige), pålidelighed, ændringsforventninger
 - Lav en faktor tabel for kravene

| Faktor | Målbarhed | Variation | Faktors betydning for interessenter | Prioritering | Risiko/hvor stort er problemet |
|----------------|--------------|-----------------------|---|--------------|--|
| Stor hastighed | Max. 10 sek. | Kan blive kortere tid | Brugere Udviklere Projektplan Arkitektur | (fx MoSCoW) | evt. % evt. 'høj', 'middel', 'lav' |
| | | | | | |

- Håndtering af faktorer
 - Løsningsforslag: Tekniske memoer
 - Faktoren
 - Løsningsbeskrivelse
 - Begrundelse
 - Løse ender
 - Alternative forsøg
 - Afvejning af faktorer (start med at kigge på dem med højest risiko)
 - Evt. ændringer af krav (aftales med rekvirenten)
 - Eksperimenter som hjælp

8.13.3. Basale principper/teknikker

- Lav kobling (lav afhængighed, gives i objektorienteret programmering ved at gøre attributterne private og give adgang gennem offentlige metoder)
- Høj samhørighed
- Afskærmende variationer (mønster)
 - En del af systemet er ustabil
 - Begrænsning af skaderne
 - Teknikker: Indkapsling, polymorfi, interface, konfigurationsfiler

8.13.4. Basal løsning

- Lagdelt arkitektur
 - Tilpasning
- Specielle forhold
 - Kommunikation opad: Observer pattern
 - Applikationslag
 - Styring af funktionalitet er kompliceret

- Ved forskellige brugergrænseflader (fx web services)

8.13.5. UP og arkitektur

UML

- Package diagrammer
- Sekvensdiagrammer
 - Bør kun laves for komplicerede funktioner

UP dokumenter

- Ikke-funktionelle i kravspecifikation (evt. i supplerende specifikation)
- SAD (Software Architecture Document)

UP faser

- Inception
 - Introduktion af ikke-funktionelle krav
 - SAD introduceres (skelettet introduceres)
 - Proof-of-concept fx ved hjælp af eksperiment
- Elaboration
 - Implementering af Skelettet
 - Yderligere check af brugbarheden
- Construction
 - Brug af arkitekturen (fyld på skelet)
- Transition
 - Slut review af SAD

8.14. Vedligeholdelse

Definition

Ændring af et softwareprodukt efter aflevering med henblik på at korrigere fejl, at forbedre performance eller andre egenskaber eller at tilpasse produktet til ændrede omgivelser.

[IEEE standard]

- Op imod/ca. 80 % af de samlede omkostninger ved udvikling af et system anvendes på vedligehold
- Loven om sigende kompleksitet (entropi)
 - Koden bliver mere og mere rodet jo længere hen i vedligeholdsperioden man kommer

8.14.1. Typer af vedligeholdelse

- Rettelser
 - Gentagende rettelser fører ofte til rodet kode
- Kravbaserede tilpasninger (største gruppe)
 - Nye love, nye forretningsgange m.v.
- Tekniskbaserede tilpasninger

- Fx ved nyt OS, DB e.l.
- Forebyggelse (af entropi) (mindste gruppe)
 - En form for hovedrengøring i koden, fx restrukturering af redundant database

8.14.2. Reduktion af omkostninger

- Genbrug af gennemtestede elementer
- Planlægning af bemanning (korrekt antal ansatte til sagen)
- Specialister i vedligeholdelse
 - Kunne se mangler i både kode og systemet, kunne skabe sig et overblik
- Ordentlig dokumentation
- Kvalitetssikring undervejs

8.14.3. Organisering

W-type (Work-type)

- Fx delt op i Analyse, Programmering
 - Hænger tæt sammen med udviklingsmåden
 - Kræver meget koordinering (negativt)
 - Man har de rigtige folk til arbejdet (positivt)

A-type (Application-type)

- Delt op i forskellige systemer
 - Typisk bemandede med tidligere udviklere
 - Tæt kendskab til systemet (positivt)
 - "Blind" for fejl, sårbar ved personaleskift (negativt)

L-type (Livscyklus)

- Udvikling og vedligehold er adskilt
 - Ofte koblet til almindelig support
 - Specialisering, hurtigere reaktion på bruger-opdagede fejl, bedre planlægning for udviklerne (positivt)
 - Hvis udviklere sidder i support er det ressource spild, ikke meget prestige i jobbet (negativt)

8.14.4. Standardforløb for vedligeholdelse

- 1 Identifikation, type fastsættelse og prioritering
- 2 Analyse af ændringsønske → estimering af pris
- 3 Gennemførelse af ændringen (hvis der er grønt lys)

8.14.5. Reverse engineering

- Processen med at analysere et system for at
 - Finde dets komponenter (hvad består det af?) og dets sammenhænge
 - Lave en repræsentation på et højere niveau (fx dokumentation)

- Alternativt: System arkæologi(!)
- Bruges når systemet skal til internt eftersyn (fx ved forebyggelse el. nyskrives (system reengineering))
- Specialværktøjer og CASE-værktøjer (til fx at lave dokumentation ud fra kildekode) kan hjælpe

8.15. Implementering

8.15.1. Overordnet

- Projekt med egen livscyklus
- Centralt mellemprodukt: Implementeringsplan
- Slut produkt: Et kørende system i organisationen
- Deltagere
 - Projektgruppe: består af udviklere (den tekniske side), brugere, driftpersonale
 - Styregruppe (laver rammer og kontrollerer udviklingen): består af ledere i berørte afdelinger + it-leder
- Centrale forhold
 - Teknisk og organisatorisk opgave
 - Start: lang tid inden udviklingen er færdig
 - Vigtigt med efterbehandling (fx evaluering)

8.15.2. Komponenter

Mennesker: berørt af det nye system

- Fx direkte brugere, indirekte brugere (folk der skal bruge oplysninger fra systemet, men ikke har hands-on), andre med forbindelse til brugere (den øvrige organisation, familier m.v.)
- Mulige påvirkninger: job ændringer, fyringer, større arbejdspress m.v.

Software: største punkt for udviklerne

- Tilpasning, udbygning, udskiftning med helt nyt system, intet eksisterende IT-system
- Tage hensyn til evt. parallelkørsel (at det nye og det gamle system i en periode begge kører), mulig tilbagevenden til gammelt system

Procedurer: arbejdsprocedurer ændrer sig

- Design af (og beslutning om) ændringer (af proceduren)
- Oplæringsbehov (undervisning)
- Supportbehov (superbrugere, helpdesk hos udvikler)

Data: eksisterende datas anvendelse

- Konvertering af data (kan være et projekt i sig selv)
- Konverteringen bør så vidt muligt foregå inden installationen af det nye program
- Ved parallelkørsel er der dataproblemer (med hensyn til persistens)

Hardware: ofte et overset problem

- Tit behov for udvidelse af maskiner/netværk

- Der kan være behov for midlertidigt udstyr/maskiner
- Opdatering/udvidelser bør være på plads inden installationen af softwaren

8.15.3. Implementationsplan

Har samme struktur som en udviklingsplan.

- Formål/mål (kan normalt overføres fra udviklingsprojektet)
- Strategi
 - Direkte overgang: stop det gamle, installer det nye, start op
 - Positivt: spare tid og penge
 - Negativt: indkørvanskeligheder, gambling (kan koste meget)
 - Parallelkørsel: installation af nyt (det gamle kører fortsat), noget tid går, luk det gamle system ned
 - Positivt: sikkerhedsventil, mister ikke data ved problemer
 - Negativt:
 - Prototype implementering: en del af systemet tages i brug, eller hele systemet tages i brug i en afdeling/mindre del af virksomheden/organisationen
 - Positivt: sikkerhedsventil, begrænset skade
 - Negativt: problemer med kommunikation med det gamle system
- Risikovurdering (ved implementering og kommende drift)
- Interessent analyse (mennesker)
- Projektmodel: vandfald (før, under, efter installation)
- Organisering (hvem gør hvad)
- Estimering af ressourceforbrug
- Faser (før, under, efter) og faselinier/milestones
 - Hvordan skal projektet se ud/hvad skal være klart/gennemført når dét punkt nås

8.15.3.1. Faser og faselinier

| Fase | Teori (bruger) | Praksis (leverandør) jf. AMJs erfaringer |
|------------------|---|---|
| Før installation | Implementeringsplan Mennesker: motivering af brugere, initiativ til oplæring Software: systemtest/accepttest (check), backup af eksisterende system Procedurer: beskrive ændret organisation og arbejdsgang Data: prøvekonvertering Hardware: sikre at tekniske krav er opfyldt, nye komponenter installeret (hvis muligt) – også basis software | Aktivitet i systemudviklingsprojektet, estimering ud fra erfaring Mennesker: 2*2 personer (rolle-/roteringsprincip) Software: test på plads (systemtest) Procedure: brugervejledning klar Data: prøvekonvertering (kopi af live database) Hardware: udsende krav til skolerne i god tid, installation af HW skulle være på plads |

| | | |
|----------------------|---|--|
| Under installationen | Backup af data (og software) Hardware og støtte software klargøres Installation af nyt program (server + klient) Initialisering/opsætning af data Konvertering af data Drift prøve | Installationsprogram: Backup tages, Check af HW/SW, server installation (passwords fra systemadministrator, support mulighed), initialisering og konvertering inkluderet Drift prøve af superbruger |
| Efter installationen | Eventuel test (fx belastningstest) Initialisering af driftsrutiner (backup, start/stop m.v.) – oplæring af systemadministrator Evaluering af implementering Sikring af at brugerne er tilfredse Initialisering af support | Accepttest – evt. tilpasning Driftsrutiner, sjældent behov for nyt Intern evaluering (2*2) inklusiv fejring Løbende support på plads (pga. tidligere versioner af systemet) |

8.16. Procesforbedring

Stammer fra pentagon (USA's militær center), de oplevede at der var stor forskel i kvaliteten af de systemer som eksterne leverandører kunne præstere.

8.16.1. Intro

- Behov for at forbedre systemudvikling fra projekt til projekt
- Modeller til dette: CMM/CMMI, Bootstrap, SPICE

8.16.2. CMMI (Capability Maturity Model Integration)

- o Anviser en vej til procesforbedring i en SW virksomhed
- o Anvendes også af kunder til check af leverandører af systemer
- o Findes også i versioner til andet end SW udvikling

8.16.2.1. Består af fem niveauer

(1) initial – begynder niveau

- Kaotisk udvikling – succes beror på enkeltpersoner
 - Ad hoc styring/brandslukning
 - Procesområder: ingen
 - o Uforudsigeligt og dårligt kontrolleret
 - o Videre: disciplineret proces
- ##### (2) repeatable – styret niveau
- Det er muligt at gentage noget, der er gået godt, hvis blot det ligner og man ikke er for presset
 - Procesområder: kravstyring, projektplanlægning, måling og analyse, projektopfølgning, konfigurationsstyring, (versionsstyring), kvalitetssikring, underleverandørstyring
 - o Kan gentage tidligere udførte opgaver

- Videre: standard proces
 - (3) defined – dokumenteret niveau
 - Styr på udviklingsprocesser generelt – overskud til at kigge på tværs af projekter
 - Procesområder: udvikling af krav, teknisk løsning, produktintegration, verifikation og validering, organisatoriske emner, risikostyring, beslutningsanalyse, og flere...
 - Mest test (og review)
 - Proces forstået
 - Videre: forudsigelig proces/projekt
 - (4) managed – kvantitativt styret niveau (målbart)
 - Fokus på målinger og dataindsamling
 - Procesområder: organisatorisk procesdygtighed, kvantitativ projektledelse
 - Proces/projekt er målt og kontrolleret
 - Videre: Forbedre proces
 - (5) optimizing – optimerende niveau
 - Fokus på kontinuerlig forbedring af processen
 - Procesområder: organisatorisk fornyelse og udbredelse, årsagsanalyse og forbedring
 - Fokus på procesforbedring
- ### 8.16.3. Opsamling
- Kræver tid og penge
 - Vigtigt at overveje ønsket slutniveau (der er ikke mange på niveau 5)
 - Kræver total opbakning i hele virksomheden
 - Omkring 40% af alle virksomheder på niveau 4 og 5 ligger i Indien
 - Tidligere har niveau 5 primært været virksomheder indenfor luft- og rumfart