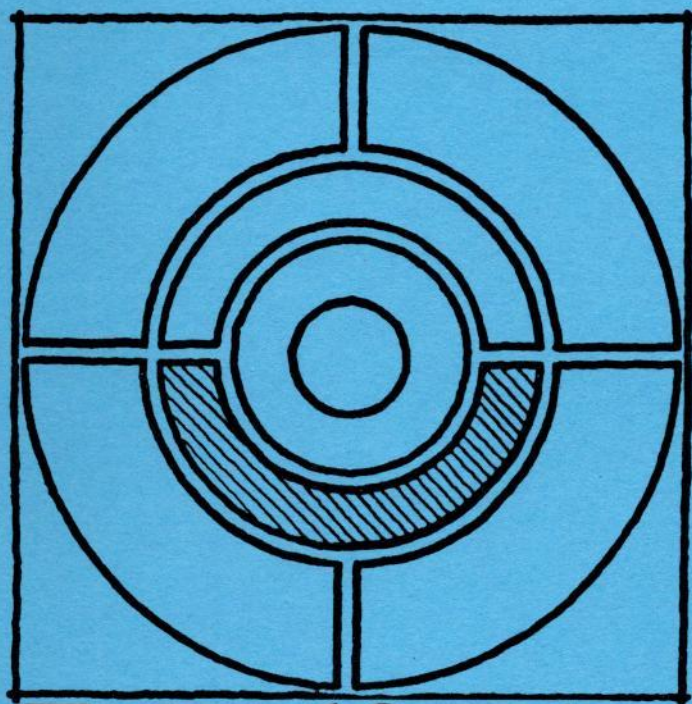




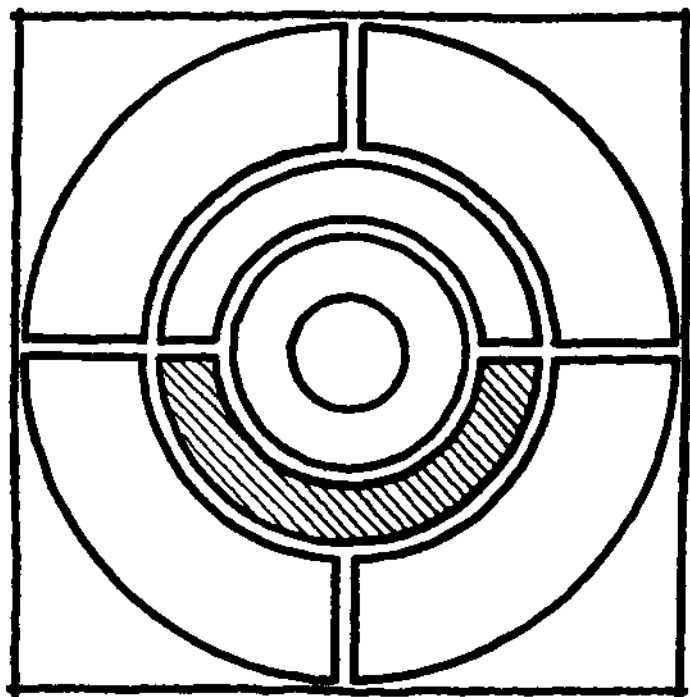
Testudførelse



Testudførelse



Testudførelse



Testudførelse

Testudførelse

© Datacentralen A/S

Version 1, november 1991

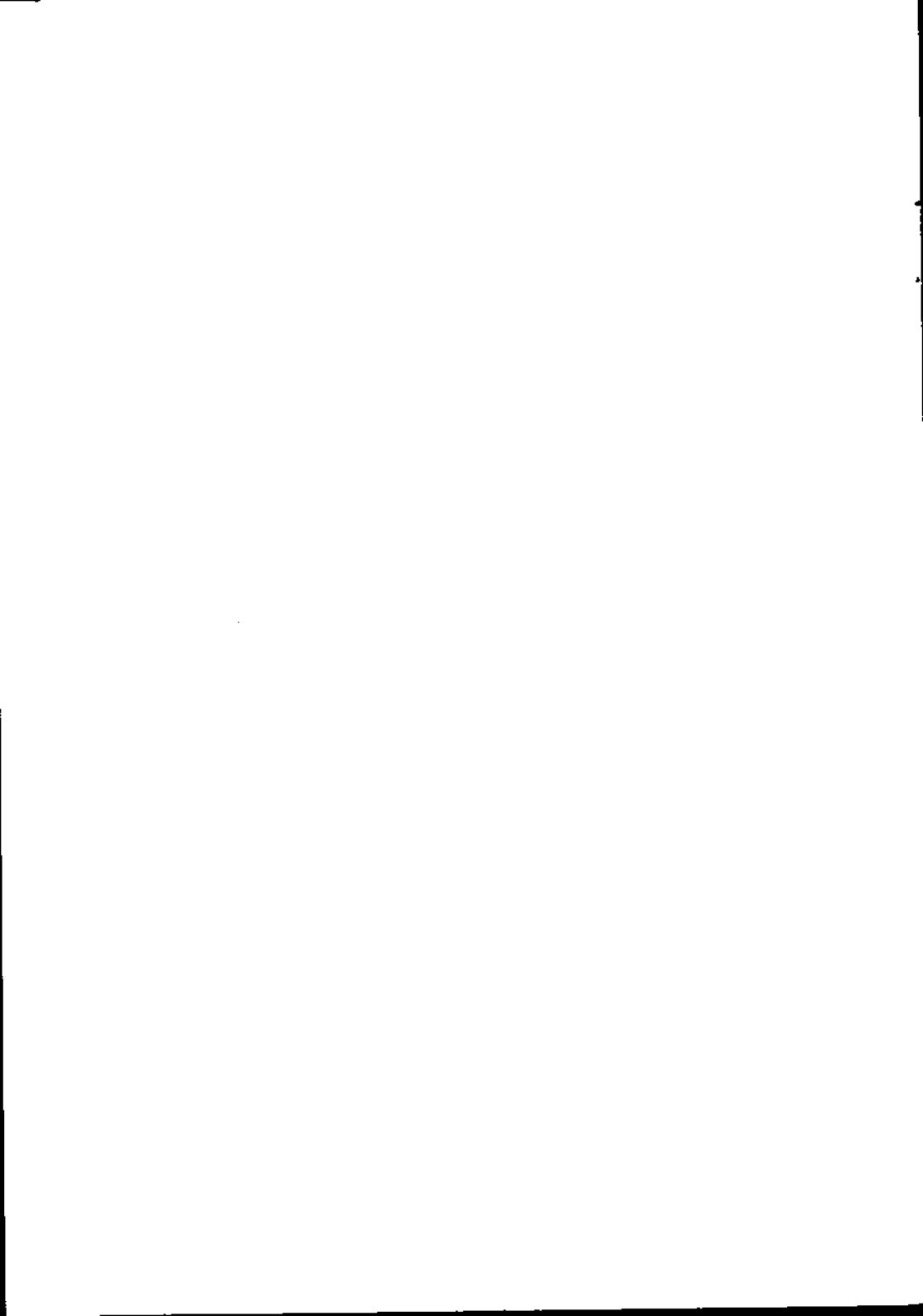
Ansvarlig:

Systemudviklingservice

Fordelingsnøgle: HTUF1

Indholdsfortegnelse

Indledning og formål	5
Udarbejdelse af testmodeller	6
Dokumentation af testmodeller	6
Testoversigt	8
Dokumentation af de valgte testeksempler	8
Teknikker til dannelse af testeksempler	10
Modultest	11
Integrationstest	13
Systemtest	14
Testafvikling	18
Klargøring af afvikling	18
Klargøring af inddata	18
Afvikling af testjob	19
Rapportering af fejl og uhensigtsmæssigheder	19
Afvikling af resultatmateriale	20
Bilag: Testspecifikation	21
Indhold	21
Testmodel 1.1.3	21
Stikords-register	26



Indledning og formål

Denne brochure er en vejledning i at udføre test af systemer.

Formålet med brochuren er at beskrive teknikker til at udføre og dokumentere en allerede planlagt intern test.

Brochuren er opdelt i følgende afsnit:

- Først beskrives, hvordan en testmodel kan udarbejdes, inddelt i et afsnit om dokumentation af testmodeller og et afsnit, der giver ideer til, hvordan testeksempler dannes.
- Derefter gennemgås de enkelte trin i testafviklingen med huskelister og praktiske anvisninger til de enkelte opgavers udførelse.

I brochurens bilag findes et eksempel på, hvordan en testspecifikation kan se ud.

Testudførelses-brochuren henvender sig hovedsagelig til testkoordinatorer og de personer, der udfører den praktiske test.

Brochuren er justeret efter de krav til test, som er en del af Datacentralens kvalitetsstyringssystem.

Udarbejdelse af testmodeller

For at andre kan gennemskue effektiviteten og kvaliteten af en testmodel, er det væsentlig, at den er dokumenteret både systematisk og gennemskueligt.

Dokumentation vedrørende testmodeller samt fejlrapporter og opnået resultatmateriale samles i testspecifikationen, som arkiveres, indtil systemets garantiperiode i følge kontrakten er udløbet. Hvis der ikke er garanti på systemet gemmes testspecifikationen mindst et år.

Dokumentation af testmodeller

For at kunne finde rundt i dokumentationen af testen for et system skal testområderne nummereres systematisk.

Figuren på næste side viser et eksempel på, hvordan nummereringen af testområderne kan foretages:

Systemtest	Integrationstest	Modultest
batch-del testområde 1	batch ugekørsel testområde 11	udtræk liste 1 testområde 116
	batch årskørsler testområde 12	udtræk årstotal testområde 125
online ajourføring testområde 2	online oprettelse testområde 21	opret stamoplysning testområde 211
	online ajourføring testområde 22	ajourfør stamopl. testområde 222
		slet stamopl. testområde 223
online forespørgsel testområde 3	online forespørgsel testområde 31	foresp. stamopl. testområde 314

Det ses, at systemtesten er opdelt i 3 testområder. Til integrationstesten nedbrydes disse områder i passende dele. Disse nye testområder nummereres med systemtestområdets nummer efterfulgt af et løbenummer. Til modultesten gentages processen for hvert af integrationstestens områder.

Det kan være hensigtsmæssigt at tilpasse testområdernes nummerering til nummereringen i systemets dataflowdiagrammer.

Testoversigt

Til hjælp ved den systematiske udvælgelse af testeksempler findes blanketten TESTOVERSIGT. Den bruges til at opstille en systematisk oversigt over samtlige mulige kombinationer af inddata og registerdata samt til at beskrive forventet resultat.

I bilaget findes et eksempel på, hvordan blanketten kan bruges.

Når denne totaloversigt er opstillet, udvælges de testeksempler, der skal med i afviklingen af testen. Herefter er det muligt at vurdere konsekvensen af ikke at udføre og afvikle de fravalgte testeksempler.

Til opstillingen af disse oversigter benyttes black-box teknikker, white-box teknikker eller en kombination af disse.

Dokumentation af de valgte testeksempler

Dokumentationen vedrørende de testeksempler, der udvælges til at deltage i afviklingen af testen, omfatter beskrivelse af:

- den regel/betingelse testeksemplet vedrører
- inddata og data på registre
- forventet resultat.

Til dokumentation af test af batch- og online-systemer benyttes principielt de samme dokumentationselementer. De benyt-

tede blanketter har dog ofte et noget forskellig udseende, hvorfor de i det følgende beskrives hver for sig.

Batch-systemer

Med hensyn til beskrivelse af den regel/betingelse, der skal testes, vil det ofte være nok at have en henvisning til den øvrige systemdokumentation, f.eks. kan der ved integrationstest henvises til systembeskrivelsen.

Til detaljeret beskrivelse af inddata, data på registre og forventet resultat kan bruges en TESTENHEDSBLANKET. Der udfærdiges en for hvert testeksempel, der medtages i testafviklingen. Alle testeksempler er tildelt en testenhed på testoversigten. Denne testenhed påføres de blanketter, der vedrører det pågældende testeksempel.

I bilaget findes et eksempel på, hvordan blanketten bruges.

Enten som supplement til eller som erstatning for testenhedsblanketten kan bruges blanketter, der udvikles af testkoordinatoren i forbindelse med planlægningen af testen.

Online-systemer

Beskrivelsen af regler og betingelser foretages på samme måde som for batch-systemer.

Til detaljeret beskrivelse af inddata, data på registre, forventet resultat og logning anvendes en af følgende teknikker afhængig af, hvor detaljeret beskrivelsen skal være:

1. Testenhedsblanketten bruges som beskrevet i forbindelse med test af batch-systemer.
2. Testkoordinator kan ofte med succes udvikle specielle blanketter, der relaterer sig direkte til systemets skærbilleder, til dokumentation af inddata og forventet resultat. I disse tilfælde bør også udvikles blanketter til data på registre og log, da beskrivelse af disse ellers ofte glemmes. Testeksempel nr. 1 i testspecifikationen i bilaget er udarbejdet efter denne teknik.
3. Der skrives en "drejebog", som viser de mest betydende data vedrørende inddata, registerdata og forventet resultat samt rækkefølge for de valgte testeksempler. Ved første afvikling af testmodellen dokumenteres inddata og faktisk resultatmateriale ved print afskærbilleder og registre.

Alle testeksempler er tildelt en testenhed på testoversigten. Denne testenhed påføres de blanketter, der vedrører det pågældende testeksempel.

Tekniker til dannelse af testeksempler

De teknikker, der benyttes i forbindelse med systemtest, integrationstest og modultest, kan ikke adskilles fuldstæn-

digt. De er i det følgende beskrevet i forbindelse med den testform, hvor de oftest ses benyttet.

Modultest

Af de teknikker, der anvendes til modultest, er det kun white-box teknikkerne og undersøgelse af registerbenyttelsen, der er specielle for modultest. Black-box teknikkerne, som beskrives i forbindelse med systemtest, skal bruges som supplement, hvis testen skal være dækkende.

Både white-box teknikkerne og undersøgelse af registerbenyttelsen kræver kendskab til modulets indre opbygning.

White-box teknikkerne

White-box teknikkerne anvendes som hjælp til at danne testeksempler, der søger efter tilfælde, hvor der er fejl i logikken. I det følgende beskrives nogle af white-box teknikkerne:

- sætningsdækning vil sige, at man udarbejder testeksempler, så hver sætning i et modul udføres mindst én gang.
- beslutningsdækning eller forgreningsdækning er, at man udarbejder testeksempler, så hver beslutning i et modul testes sand og falsk mindst én gang. Dette medfører, at enhver handling udføres mindst én gang.

- betingelsesdækning vil sige, at man udarbejder testeksempler, så hver betingelse i en beslutning skal give alle de mulige resultater mindst én gang, og hver handling udføres mindst én gang.
- Beslutnings-/betingelsesdækning er, at man udarbejder testeksempler, så hver betingelse i en beslutning antager alle mulige resultater mindst en gang, hver beslutning antager alle mulige udfald mindst én gang, og alle handlinger udføres mindst én gang.
- Flerbetingelsesdækning vil sige, at man udarbejder testeksempler, så alle kombinationer af betingede resultater i hver beslutning og alle handlinger aktiveres mindst én gang. Det kræver en stor mængde testeksempler.

Undersøgelse af registerbenyttelsen

Denne form for test er kun aktuel, hvis modulet henter eller opdaterer data via et databasesystem.

Testen har til formål at afsløre, om databasesystemet finder frem til de ønskede data på den mest hensigtsmæssige måde. Det vil sige en test af, om database design og brugen af data passer sammen.

Hvis man foretager denne test, gøres det ved at udføre et sæt testeksempler, hvor modulets databasekald betingelsesdækkes. Der skal udføres testeksempler til afvikling på både en nyop-

rettet database og en database, der har været ajourført flere gange med både oprettelser, ajourføringer og sletninger.

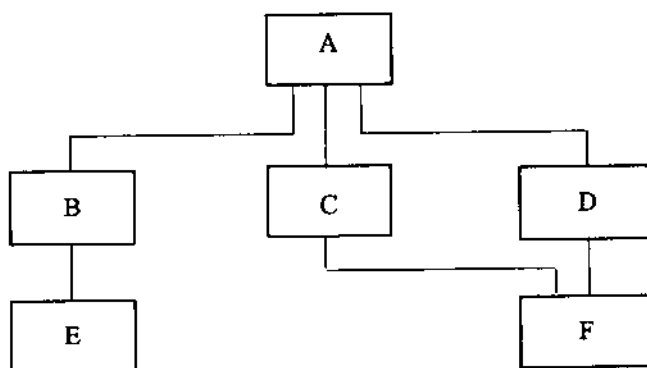
Afviklingen af disse specielle testeksempler overvåges med et værktøj, der kan klarlægge, hvilke veje databasesystemet benytter for at finde frem til de ønskede data.

Når et system bruger registre, hvor data hentes via index undersøges på ligende måde, om registret opbygges hensigtsmæssig, når det opdateres.

Integrationstest

Der findes ingen teknikker specifikt til dannelse af testeksempler til integrationstest. Det nærmeste man kommer, er en anvendelse af black-box teknikkerne i kombination med, at følgende områder dækkes af testeksempler:

- alle inddata og uddata fra hvert modul undersøges. Det vil sige, at et modul, der kan kaldes med forskellige parametre, skal testes ligeså mange gange, som dets parametre kan kombineres
- alle moduler og kald afprøves i kombination. Til systemet på omstående figur laves testeksempler, så modul F kaldes fra både modul C og D
- standardmoduler testes på samme måde som systemets egne moduler.



Systemtest

Ved systemtest forventes intet kendskab til systemets indre struktur. Det er alene en undersøgelse af, at systemet opfylder de stillede krav.

Til dannelse af testeksempler skal derfor anvendes teknikker, der ikke kræver kendskab til systemets indre struktur. Black-box teknikkerne er egnede til formålet.

Black-box teknikkerne

Der vælges mellem følgende tre black-box teknikker eller en kombination af dem:

- opdeling i ækvivalensklasser
- grænseværdianalyser
- erfaringsbetinget test.

Opdeling i ækvivalensklasser

Ved opdeling i ækvivalensklasser indsnævres mængden af nødvendige testeksempler ved, at man inddeler inddata i grupper, hvor systemet forventes at reagere ens på alle værdier indenfor en given gruppe.

Opdeling i ækvivalensklasser skal gøres for én regel eller betingelse ad gangen. Man vil typisk foretage en sådan opdeling i forbindelse med test af regler/betingelser, hvor mange forskellige værdier giver få forskellige resultater.

Man skelner mellem gyldige og ugyldige ækvivalensklasser. Værdierne i en gyldig ækvivalensklasse accepteres af systemet, mens ugyldige ækvivalensklasser afvises/giver fejludskrift.

Som eksempel kan nævnes en test af en betingelse i en kravspecifikation, hvor en vares salgspris kan ligge mellem 5 kr. og 500 kr. Ækvivalensklasserne for denne salgspris vil være:

- værdier mindre end 5 (ugyldig ækvivalensklasse)
- værdier større end 4 og mindre en 501 (gyldig ækvivalensklasse)
- værdier større end 500 (ugyldig ækvivalensklasse).

Grænseværdianalyser

Grænseværdianalyser kan anvendes som udgangspunkt for dannelse af testeksempler, hvor inddataværdierne ligger lige på-, lige under eller lige over grænsen.

Hvis en inddatabetingelse specificerer et interval af værdier, skal der dannes gyldige testeksempler med værdier netop på grænserne og indenfor grænserne. Desuden skal der dannes ugyldige testeksempler med værdier præcis uden for grænserne.

Eksempel:

En inddataværdi har et gyldighedsområde fra -1 til 1 med tre decimaler. Der skal dannes testeksempler med inddataværdier på -1,000 og 1,000 og på -1,001 og 1,001 og på -0,999 og 0,999.

Hvis en inddatabetingelse specificerer et antal (en mængde), skal der dannes gyldige testeksempler med minimumsantal og maximumsantal. Desuden skal der dannes ugyldige testeksempler med antal præcis under og over disse værdier.

Eksempel:

Et system har en inddatafil, der kun kan indeholde 1-255 poster. Der skal i dette tilfælde dannes testeksempler, der indeholder 0, 1, 255 og 256 poster.

Hvis der findes krav vedr. uddatatilstande, skal det afprøves, om kravene er opfyldt.

Eksempel:

I et informationssøgningssystem er der krav om, at der som maximum må vises 4 uddrag.

Der skal i dette tilfælde dannes testeksempler, som forårsager, at der vises nul og fire uddrag. Desuden skal der dannes et testeksempel, der kan vise, om der fejlagtigt vises fem uddrag.

Erfaringsbetinget test

Mange mennesker har et vist naturtalent for at "opsnuse fejl". De anvender en mere eller mindre bevidst teknik, som er baseret på tidligere erfaringer, til hjælp til udformning af testeksempler. De har både gennem intuition og erfaring en anelse om, hvilke fejl der er sandsynlige, og de udarbejder derefter testeksempler, der kan afsløre disse fejl.

Denne viden skal blandt andet anvendes, når man skal vælge de testeksempler, som skal udarbejdes i praksis og deltage i testafviklingen.

Her er en fejl-erfaringsliste, som angiver områder, hvor fejl let kan opstå:

- datoer (års-, årti-, århundrede- årtusindeskifte)
- tabeller (tomme tabeller eller tabeller med overflow)
- end-of-file
- helt normale situationer
- gentagelse af ens situationer
- kombinationer af regler
- regler hvor og/eller indgår
- brug af æ, ø og å
- sorteringer.

Testafvikling

I forbindelse med afviklingen af testen skal varetages følgende arbejdsopgaver:

- klargøring af afvikling
- klargøring af inddata
- afvikling af testjob
- rapportering af fejl og uhensigtsmæssigheder
- arkivering af resultatmateriale.

Klargøring af afvikling

Uanset hvilket basissystem, man arbejder med, skal der altid laves procedurer til at starte og sammenbinde systemets moduler. I forbindelse med testen skal man specielt være opmærksom på at få de testværktøjer med, som testplanen beskriver, der skal bruges.

Man skal også være opmærksom på, at der ofte kræves ekstra back-up af database og registre for at kunne genskabe udgangspunktet for en test, efter at der er fundet fejl, og testen skal afvikles igen.

Klargøring af inddata

Klargøring af inddata er vigtigt i forbindelse med test af batch-systemer, hvor inddata skal findes på maskinlæsbar

medie, for at systemets moduler kan bruge dataene.

Afvikling af testjob

I forbindelse med test kan afvikles følgende typer af job:

- simulering af produktion
- oprettelse af testregistre
- genskabelse af udgangspunktet for testen
- job, der skaber udgangspunkt for situationer, der ikke kan opnås gennem almindelig driftssituation, f.eks når det i forbindelse med modul-test skal undersøges, hvordan et modul reagerer på, at der mangler data i en database, som ifølge databasens design ikke kan mangle
- back-up kørsler.

Rapportering af fejl og uhensigtsmæssigheder

Opdagede fejl og uhensigtsmæssigheder kan rapporteres mere eller mindre formelt. Det er normalt, at fejlrapporteringen er mindst formel i modultesten og mest formel i systemtesten.

I testplanen er fastlagt, hvordan fejlrapporteringen skal foregå inden for det enkelte testområde.

Den eneste formelle registrering af fejl og u hensigtsmæssigheder er fejlrapporten. Udfyldte fejlrapporter afleveres til testkoordinator, som kan registrerer rapportens formelle oplysninger i et enten manuelt eller maskinelt fejlrapportstyrings-system.

Hvis den enkelte rapports status til stadighed indberettes til styringssystemet, kan testkoordinator hele tiden følge, hvor langt man er i testen.

Når fejl og u hensigtsmæssigheder håndteres formelt, er det lettere at afgøre, hvornår test kan afsluttes, da man hele tiden kan lave statistikker, der viser:

- hvor mange fejlrapporter der er udfyldt og vedrørende hvilke moduler
- hvor mange fejl der er fundet indtil et givet tidspunkt
- hvilken status fejlrapporterne har.

Arkivering af resultatmateriale

Når testen af et område er afsluttet, arkiveres det sidste sæt resultatmateriale. Det kan eventuelt bruges til sammenligning med resultatmaterialet fra en senere gentagelse af testen.

Bilag: Testspecifikation

Indhold

Testmodel 1.1.1

Testmodel 1.1.2

Testmodel 1.1.3

Formål

Testoversigt

Udførte testeksempler med forventet resultat

Opnåede resultater

Testmodel 1.1.4

.....

Testmodel 1.1.3

Formål

At afteste forespørgselsdelen af bestilling af kørsler som beskrevet i BR specifikation 1.1.2, FKØR.

Testoversigt se blanketbillede på næste side

Testeksempel nr. 1

Testeksemplet undersøger, at der vises 5 kørsler, når der findes fem kørselsdatoer på DRPLKOER. Forventet resultat er et normalt billede.

Inddata

Forespørgsel vedr. kørsler	Testeksempel nr.: 1
Inddata	Testenhed: 101
SYSTEM:BR BILLEDE:FKØR KMNR: 101 DELSYSTEM: P ÅR: 91	

Testenhedsblanket se blanketbillede næste side

Forventet resultat

Forespørgsel vedr. kørsler

Testeksempel nr.: 1

Forventet resultat

Testenhed: 101

Kmnr: 101 Navn : København

Sidste opdatering : 22 10 1991

Kontaktperson : Anne Jensen

Valg af kørsler

1. kørsel : x

2. kørsel : x

3. kørsel : x

4. kørsel : x

5. kørsel : x

SYSTEM:BR BILLEDE: FKØR KMNR: 101 DELSYSTEM: P ÅR: 91

Stikordsregister

A

Afvikling 19

B

Batch-system 8, 9, 18

Beslutningsdækning 11

Betingelsesdækning 12

Black-box teknik 8, 11, 13, 14

D

Database 12, 19

Dataflowdiagram 7

Dokumentation 6, 8

E

Erfaringsbetinget test 14, 17

F

Fejl-erfaringsliste 17

Fejlrapport 19

Flerbetingelsesdækning 12

Forgreningsdækning 11

Forventet resultat 8, 9, 10, 21, 25

G

Garantiperiode 6

Grænseværdianalyse 14, 15

I

Integrationstest 7, 9, 10, 13

K

Kontrakten 6

L

Logning 9, 10

M

Modultest 7, 10, 11, 19

O

Online-system 8, 9

P

Planlægning af test 9

R

Rapportering af fejl 18, 19

Registerbenyttelse 11, 12

Registerindhold 8, 9, 10

Resultatmateriale 6, 10, 18, 20

S

Systembeskrivelse 9

Systemtest 7, 10, 11, 14, 19

Sætningsdækning 11

T

Testafvikling 9, 17, 18

Testeksempler 8, 9, 10, 11, 12, 13, 14, 15, 17, 21

Testenhed 9, 10, 23
Testenhedsblanket 9, 10, 23
Testkoordinatoren 9, 10, 20
Testmodel 6, 21
Testområde 6, 7, 20
Testoversigt 8, 9, 10, 21
Testplan 18, 20
Testregistre 19
Testspecifikation 6, 10

W

White-box teknik 8, 11

Æ

Ækvivalensklasser 14, 15

DataCentralen