

Bilag 7a

Testhåndbog

Skal ikke udfyldes af Tilbudsgiver i forbindelse med afgivelse af tilbud.

KAKI = Cyber Divisionen, Forsvarsministeriets Materiel- og Indkøbsstyrelse.



KAKI TESTHÅNDBOG



Indholdsfortegnelse

1. Indledning	4
1.1 Overordnede krav til test	4
1.2 Oversigt over forkortelser	5
1.3 Principper for test	5
1.4 Organisering og ansvar	6
2. Testniveauer	7
2.1 Testtyper pr. testniveau	8
2.1.1 Definition af testtyper	9
2.1.2 Ansvarsfordeling og testmiljø	9
2.1.3 Testforløb fra udvikling til test – traditionel tilgang	10
2.1.4 Testforløb ved agil leverancemodel	12
3. Testprocessen	17
3.1.1 Testleverancer	17
3.1.2 Testaktiviteter	19
3.1.3 Vedligeholdelse af test og regressionstest	21
3.1.4 Roller og ansvar i forbindelse med test	21
3.1.5 Start- og slutkriterier for test	22
4. Testdækning	23
4.1 Testens scope	23
4.1.1 Test af systemets egenskaber	26
4.2 Testmetoder og teknikker	28
4.2.1 Sporbarhed til Krav	28
4.2.2 Use case baseret test	28
4.2.3 Risikobaseret test	29
4.2.4 Tidlig test og samarbejde om test	29
4.2.5 Statisk test	29
4.3 Testdata	29
5. Testværktøj og procedurer	31
5.1 Defektkategorier og defektprocedure	32
6. Dokumentformalia	34
6.1 Dokumentplacering	34
6.2 Procedure for opdatering	34
6.3 Godkendelse af fagligt indhold	34
Bilag 1	Testprocedure for IKT-projekter
Bilag 2	Testprocedure for DeMars



Forord

Denne Testhåndbog (THB) er gældende standard for test af IT-leverancer i KAKI.

Håndbogen beskriver principper for test, testmodel og de begreber der anvendes samt kravene til test og til testprocessen. Den konkrete testprocedure kan variere i forskellige dele af organisationen og beskrives i særskilte bilag til THB.

Testelementet er ansvarligt for udfærdigelse og vedligeholdelse af håndbogen. Alle henvendelser om THB rettes via FIIN til FMI-KTP-KI-DV-AFTESTNING.



1. Indledning

KAKI ønsker at levere IT løsninger og ydelser med den fornødne og aftalte kvalitet.

Test indgår som en væsentlig del af kvalitetssikringen.

KAKI Testhåndbogen skal anvendes som generelle retningslinjer for test, som skal tilpasses den enkelte leverance.

Tilpasning af testindsatsen foretages ud fra flg. principper:

- Balanceret tilgang
 - Testtyper tilvælges ud fra leverancens karakter
 - Testens indhold og omfang bestemmes ud fra en risikovurdering
 - Omfang af test tilpasses ud fra kravene til leverancen
- Formål med at teste
 - Test skal verificere, at eksisterende løsninger/ydelser fortsat kan leveres med aftalt kvalitet, når ændringer eller nye løsninger leveres
 - Test skal verificere, at kravene til leverancen er opfyldt
 - Test skal finde fejl inden idriftsættelse af leverancen, sådan at risikoen for at der findes betydende fejl i produktion minimeres
- Best practice
 - Andre testværktøjer og metodikker kan anvendes af eksterne leverandører, når det er aftalt og anses for indenfor 'best practice' for test af IT leverancer

1.1 Overordnede krav til test

I forbindelse med nye IT systemleverancer og ændringer til de eksisterende IT systemer skal testansvarlige fra Forsvarsministeriets koncern (herefter benævnt koncernen) og leverandørerne i samarbejde planlægge og gennemføre testforløb som beskrevet i denne testhåndbog.

Test skal gennemføres i alle nedenstående tilfælde, uanset om de er udført i relation til drift, vedligehold eller udvikling:

- Ændring af eksisterende IT systemers funktionalitet og grænseflader
- Fejlrettelser til eksisterende IT systemer
- Etablering af nye IT løsninger
- Installation af patches og opgradering til nye versioner
- Ændring af IT infrastruktur og basiskonfigurationer, der anvendes af eksisterende IT systemer

ISTQB standarden for test anvendes, så det sikres, at de anvendte begreber er veldefinerede og best practice indenfor test følges og herudover anvendes standarden for testdokumentation IEEE29119-3.

Principperne i THB skal følges, men der skal foretages en styret og dokumenteret skalering af testen, sådan at den svarer til opgavens omfang og karakter.

Testprincipperne er gældende både for test ved traditionelle metoder og ved agile metoder. Agil test er beskrevet i et særskilt afsnit.



1.2 Oversigt over forkortelser

Oversigt over brugte forkortelser i THB.

Forkortelse	Betydning
ALM/QC	Application Lifecycle Management/Quality Center
FIIN	Forsvarets Interne Informatik Netværk
FA	Forretningsansvarlig
IKT	Informations- og kommunikationsteknologi
ISTQB	International Software Testing Qualifications Board
IEEE29119-3	ISO standard for software test dokumentation
ITA	IT-ansvarlig
LEV	Leverandør
PGH	Programmeringshåndbog
THB	Testhåndbog
PHB	Projekthåndbog
E2E	Ende til ende

1.3 Principper for test

Formålet med gennemførelse af test er at sikre, at leverancen overholder de specificerede krav og at fejl findes og rettes tidligst muligt, inden leverancen skal i produktion og ibrugtages. Herudover at der ikke er opstået fejl i den/de eksisterende løsninger som følge af leverancen.

Der anvendes følgende 7 grundlæggende testprincipper for test:

- 1) Tidlig test.
Test gennemføres så snart det er muligt og løbende. Baggrunden for dette er bl.a., at fejl og misforståelser der findes sent i forløbet giver længere tilbageløb.
- 2) Struktureret test.
Test skal planlægges, specificeres og dokumenteres i de anvendte skabeloner og værktøjer.
- 3) Aftalt testgrundlag.
Test skal baseres på de for leverancen aftalte krav og specifikationer.
- 4) Risikobaseret test.
Testindsatsen skal prioriteres, så det sikres, at det vigtigste testes først på baggrund af leverancens risikoprofil.
- 5) Adskilte testforløb.
Test gennemføres i adskilte forløb Dvs. der skal udarbejdes testplan og testrapport pr. testforløb og de enkelte testforløb udføres adskilt.
- 6) Sporbarhed til krav.
Testen skal have sporbarhed til kravene og test-krav dækning skal dokumenteres i det anvendte værktøj.



7) Testens dækning.

Testen skal dække krav, specifikation og design i en trinvis test af leverancen. Testen skal inddrage regressionstest til dækning af eksisterende løsninger, som berøres af leverancen.

1.4 Organisering og ansvar

Retningslinjer for test udgives af KAKI Testelement, der er organisatorisk placeret i Forretningsområde DeMars. Elementet er tillige ansvarligt for uddannelse og rådgivning indenfor test samt ejer og vedligeholder KAKI testværktøjer.

Drift og vedligehold

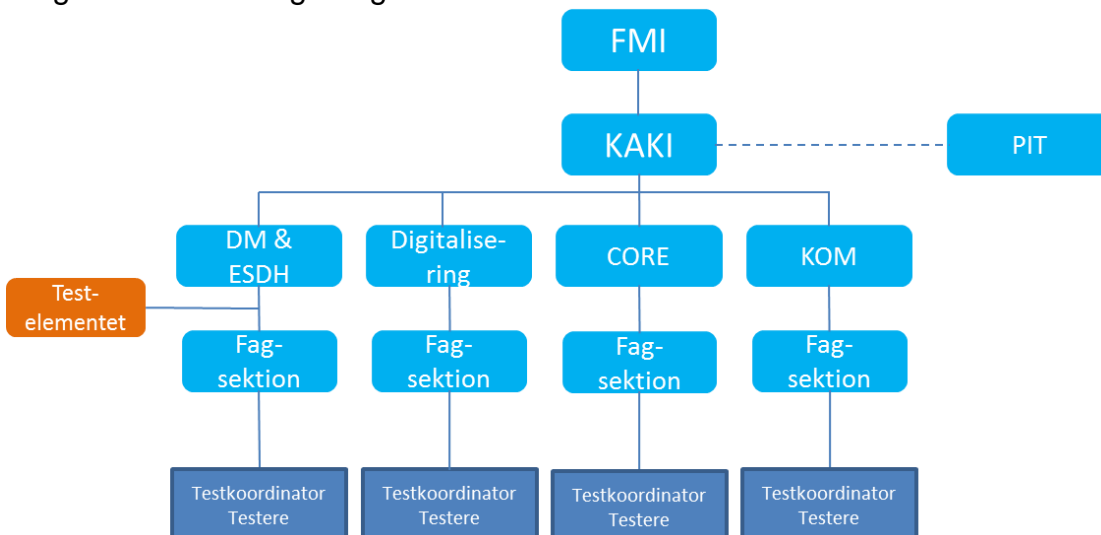
Den for et IT-system ansvarlige afdeling/(fag-)sektion er ansvarlig for, at test planlægges og gennemføres iht. denne håndbog. Test udføres af de i sektionen/afdelingen udpegede testkoordinatorer og testere. Test gennemføres i samarbejde med forretningen/kunden.

Udvikling/projekter

Nye IT-systemer eller væsentlige ændringer til eksisterende systemer leveres normalt af projekter (eller projektlignende organiseringer).

Projektlederen er ansvarlig for, at test planlægges og gennemføres iht. denne håndbog. I projektet udpeges en testkoordinator, der er ansvarlig for planlægning, afvikling og dokumentation af test.

Test udføres af testressourcer tilknyttet projektet, herunder personale fra fagsektionen, brugere fra forretningen og evt. eksterne leverandører.

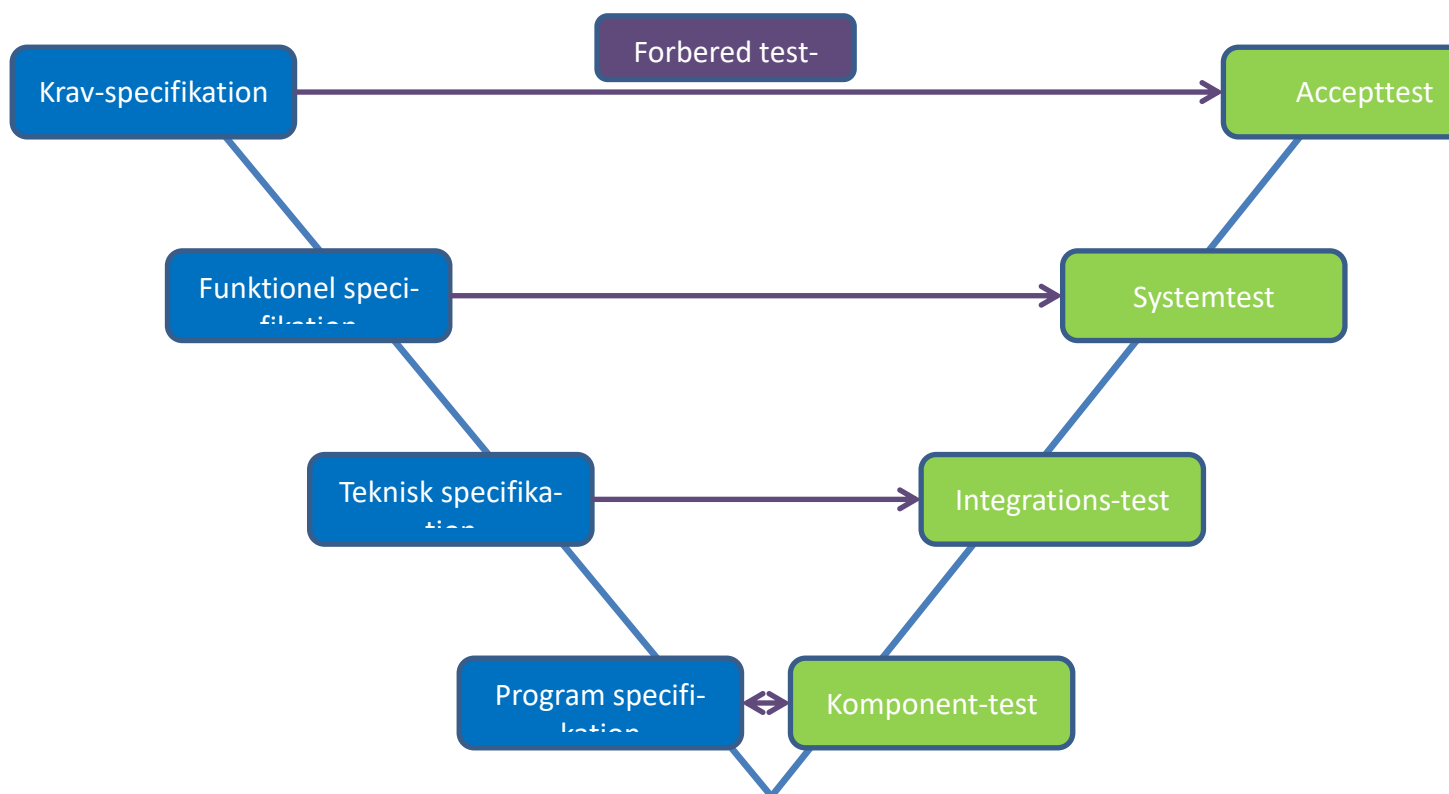


Figur 1: Testelementets placering i KAKI organisationen



2. Testniveauer

Testtilgangen i KAKI er struktureret test og baseret på V-modellen for test:



Figur 2: V-modellen for test

Anvendelse af V-modellen for test betyder, at udarbejdelse af testmateriale skal indledes parallelt med udarbejdelse af specifikation af løsningen:

SIPMO fase	Change Management fase	Specifikation	Testniveau	Testmateriale***
Ide	Planlægning*	Kravspecifikation	Accepttest	Identificere test af forretningsmål
Analyse	Planlægning*	Kravspecifikation	Accepttest	Beskrive overordnet teststrategi
Anskaffelse	Planlægning*	Kravspecifikation	Accepttest	Fastlægge teststrategi Definere acceptkriterier for krav Identificere E2E forretnings-scenarier, der skal testes
Gennemførelse	Udvikling	Funktional specifikation	Systemtest	Identificere test cases for forretningsprocesser og øvrig funktionalitet
Gennemførelse	Udvikling	Teknisk specifikation	Integrations-test	Identificere test cases for systemets interne og eks-



				terne integrationer
Gennemførelse	Udvikling	Programspecifikation	Komponenttest**	Identificere test cases for systemets komponenter
Realisering	Idriftsættelse*	Kravspecifikation (SLA)	Accepttest	Definere acceptkriterier for SLA krav

*) Ved vedligehold vurderes om ændringens størrelse, giver anledning til udarbejdelse af teststrategi, test af forretningsmål og test af SLA.

**) Under komponenttesten kan det være en fordel først at identificere test cases og umiddelbart efterfølgende gennemføre testen, sådan dette bliver et iterativt forløb imellem test og færdiggørelse af programmeringen/opsætningen af systemet.

***) Skabelon se afsnit 3.1.1

Det er særlig vigtigt, at definere acceptkriterier for krav, da dette både giver udviklingsteamet forståelse for løsningen og uddyber forretningens egne forventninger til løsningen og derved minimerer risikoen for tilbageløb i accepttesten. Da det er i denne fase, der er længst imellem specifikation og test, er det her de største tilbageløb vil være, hvis der findes fejl.

V-modellen betyder, at testen gennemføres

- På et specificeret grundlag.
- Trinvist
- Fejl identificeres og rettes på det niveau, de er introduceret.

Anvendelse af V-modellen skal således sikre, at der gennemføres adskilte testforløb og at der ikke videreføres fejl til de næste testniveauer.

2.1 Testtyper pr. testniveau

Der gennemføres følgende testforløb pr. testniveau. Nogle testtyper er ikke obligatoriske, men gennemføres, såfremt det er relevant for leverancen. Dette dokumenteres i teststrategien. Skemaet angiver ikke nødvendigvis rækkefølgen af testtyperne.



Testniveau	Testtype, der anvendes i koncernen	Obligatorisk
Installation	Installationstest	Ja hvis relevant
Komponenttest	Unittest	Ja hvis relevant
Integrationstest	Unitintegrationstest	Ja hvis relevant
	Systemintegrationstest	Ja hvis relevant
Systemtest	Smoketest	Ja
	Systemtest	Ja hvis relevant
	Konverteringstest	Ja hvis relevant
	Performancetest	Ja hvis relevant
	Sikkerhedstest	Ja hvis relevant
	Regressionstest	Ja
Accepttest	Forretningsaccepttest	Ja
	Driftstest	Ja hvis relevant

2.1.1 Definition af testtyper

Standard definition af de anvendte testtyper på overordnet niveau (ISTQB version 2.1):

Testtype	Beskrivelse
Installationstest	Test af installation af leverance jf. installationsvejledning
Unittest	Test af individuelle delkomponenter. (Kaldes også komponenttest)
Unitintegrationstest	Test af grænseflader (snitflader) og samspil mellem integrerende komponenter. (Kaldes også komponentintegrationstest)
Systemintegrationstest	Test af integration af systemer og pakker, test af grænseflader til eksterne systemer.
Smoketest	Test af at leverancen fungerer i det testmiljø, hvortil den er leveret og er klar til det videre testforløb
Systemtest	Test af det integrerede system for at verificere, at det fungerer i henhold til specifikationen og der ikke er fejl heri
Konverteringstest	Test af konvertering af data fra eksisterende systemer til brug i erstatningssystemer (data der konverteres til nye/andre systemer)
Performancetest	Test for at fastlægge et systems performance (effektivitet)
Sikkerhedstest	Test for at fastlægge hvor sikkert systemet er og om kravene til sikkerhed er opfyldte.
Regressionstest	Test af et tidligere testet program (system) efter modificering for at sikre, at defekter ikke er tilført eller afdækket i uændrede dele af softwaren som følge af de gennemførte ændringer
Forretningsaccepttest	Test i forhold til brugerbehov, krav og forretningsprocesser for at fastlægge om acceptkriterierne er opfyldte. Testen muliggør, at forretningen kan afslutte, om systemet kan accepteres
Driftstest	Testen evaluerer leverancen i dens driftsmiljø.

2.1.2 Ansvarsfordeling og testmiljø

Overordnet ansvarsfordeling af de forskellige testtyper. Leverandøren kan være en eksternt systemleverandør, men opgaver leveres også internt af KAKI (forretningsområder og



sektioner), hvorfor Leverandøren også kan være en intern Leverandør. Der kan ligeledes være flere leverandører af den samlede leverance.

Testen skal gennemføres på et testmiljø, som er egnet til testformålet med hensyn til infrastruktur, konfiguration og version af systemet og data.

Testtype	Ansvar*	Udførende	Miljø
Installation	ITA	ITA	Alle
Unittest	LEV	LEV	Udvikling
Unitintegrationstest	LEV	LEV	Udvikling
Smoketest	LEV	LEV	Test
Systemintegrationstest	FA og ITA	LEV, ITA og FA (i koordination)	Test
Performancetest	ITA	ITA og LEV (i koordination)	Test**
Sikkerhedstest	ITA	ITA og/eller LEV (afh. af kontrakt)	Test Produktion***
Konverteringstest	LEV	LEV, ITA og FA (i koordination)	Test
Systemtest	ITA	ITA	Test
Forretningsaccepttest	FA	FA	Test
Regressionstest	LEV eller ITA	LEV eller ITA	Udvikling, Test eller Staging
Driftstest	LEV	LEV (drift) og ITA (i koordination)	Produktion

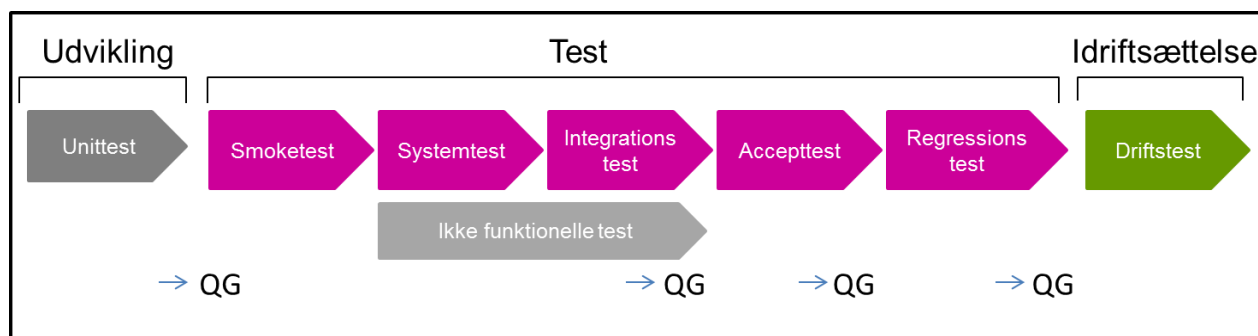
*) Ansvar for test kan være kontraktuelt fastlagt og vil i sådanne tilfælde være gældende.

**) Der vil være særlige krav til testmiljø, hvor der skal gennemføres performancetest.

***) Sikkerhedstest kan omfatte en penetration test (test hvor systemet forsøges hacket), som udføres i produktion/produktionslignende miljø.

2.1.3 Testforløb fra udvikling til test – traditionel tilgang

Et testforløb, i en traditionel tilgang, gennemføres overordnet som skitseret i figurene nedenfor for hhv. et udviklingsprojekt, vedligeholdelse og drift samt ved idriftsættelse af løsning som modtages fra eksternt leverandør eller et standard produkt. Der kan derudover være forløb som er et mix af disse f.eks. et udviklingsprojekt, hvor der indgår et standard produkt.



Figur 3: Testforløb udviklingsprojekt

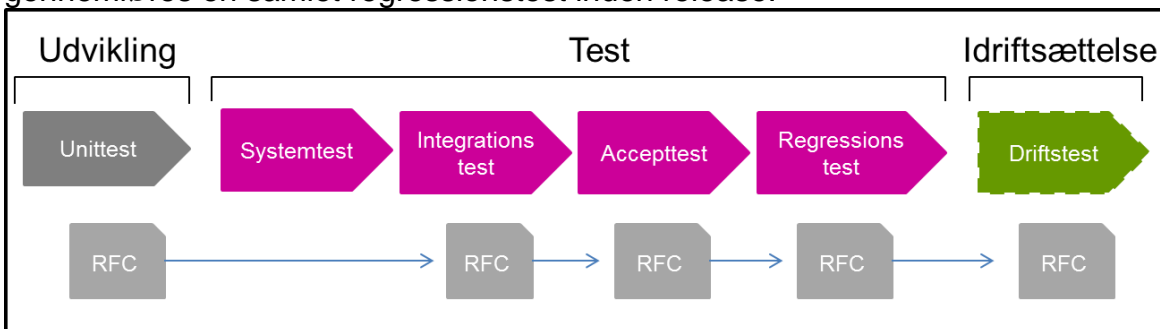


Ved den traditionelle testtilgang gennemføres og rapporteres unittest for den samlede løsning inden løsningen frigives til test. Der gennemføres en smoketest, hvorefter de enkelte testfaser gennemføres. Den enkelte testfase afsluttes inden den næste startes og opdeles evt på flere testmiljøer. Løsningen gennemgår en række quality gates imellem de enkelte testfaser hvor de aftalte kvalitetskriterier evalueres og som skal være opfyldte inden næste fase kan påbegyndes. De ikke funktionelle test gennemføres delvist parallelt f.eks. i separate testmiljøer.

2.1.3.1 Testtilgang ved vedligehold og drift

Ved vedligehold og drift testes de enkelte RFC'er og en RFC gennemløber alle testfaser fra unittest til accepttest og frigives således fra udvikling og unittest til test i den relevante fagsektion og herefter til accepttest hos forretningen inden den idriftsættes eller indgår i en release. Før en RFC kan idriftsættes skal den igennem en godkendelse (CAB) hvor kvalitetskriterier evalueres.

Ved egentlige releases af større mængde RFC'er eller ved opgraderinger mv. gennemføres en samlet regressionstest inden release.



Figur 4: Testforløb ved vedligehold og drift

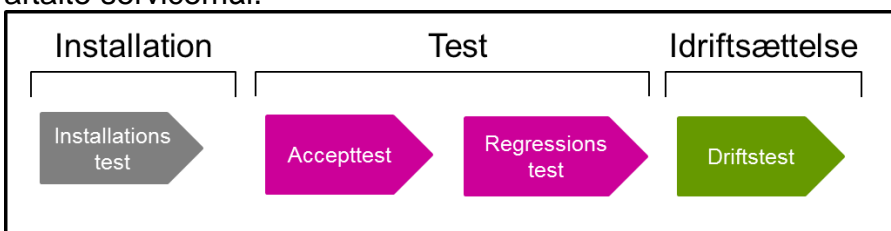
Efter idriftsættelse gennemføres driftstest som aftalt til rapportering af servicemålenes fortsatte opfyldelse.

2.1.3.2 Testtilgang ved modtagelse af leverance fra ekstern leverandør

Ved modtagelse af en løsning fra en ekstern leverandør, skal der foreligge dokumentation for leverandørens test og ved modtagelse foretages en installationstest ved hjælp af den leverede installationsvejledning i samarbejde med leverandøren.

Den relevante fagsektion skal gennemføre en accepttest for at verificere at kravene til leverancen er opfyldte samt en regressionstest der sikrer, at løsningen ikke medfører fejl i andre løsninger.

Driftstest gennemføres efter idriftsættelse og skal verificere at løsningen overholder de aftalte servicemål.



Figur 5 Test ved modtagelse af løsning fra ekstern leverandør



En løsning eller produkt der modtages fra en ekstern leverandør kan være en del af en større leverance og skal i dette tilfælde indgå i den samlede leverances teststrategi.

2.1.4 Testforløb ved agil leverancemodel

Leverancer, der leveres efter agile metoder, skal gennemføre de samme testtyper - men testen tilrettelægges indenfor de enkelte 'sprint'. Testdokumentation i form af testplan, test cases og testresultater skal udarbejdes pr sprint og testrapportering udarbejdes pr sprint som del af rapporteringen af sprintresultat ved sprintafslutning.

I en agil leverancemodel leveres der iterativt og leverancen testes og idriftsættes for hvert 'sprint'. Den agile metode mitigerer de risici, der er forbundet med, at den samlede leverance først leveres til test, efter alt er færdigudviklet og kan bedre håndtere ændringer til krav.

Testen følger således det enkelte 'sprint'. Test er en integreret aktivitet under udviklingen og testerne er en del af det agile team. Alle relevante test gennemføres for hvert sprint. Ved agil test er der 3 helt centrale elementer:

- Samarbejde i ét team – Der gennemføres test kontinuerligt under et sprint.
 - Der testes og udvikles parallelt og overdrages løbende viden, afklaringer og feedback på leverancen imellem forretning, udviklere og testere.
- Regressionstest – skal sikre en kontinuerlig integration af leverancen.
 - Der skal opbygges en regressionstestsuite der kan sikre, at fejl i den del af leverancen, som allerede er testet, findes tidligst muligt, og at der ikke opstår fejl i det eksisterende system ved ændringer.
- Automatiseret test
 - Automatiseret regressionstest kan gennemføres hurtigere og sikrer at sprint kan afsluttes uden test bliver flaskehals
 - Automatiseret regressionstest kan gennemføres kontinuerligt og fejl findes tidligst muligt ved ændringer

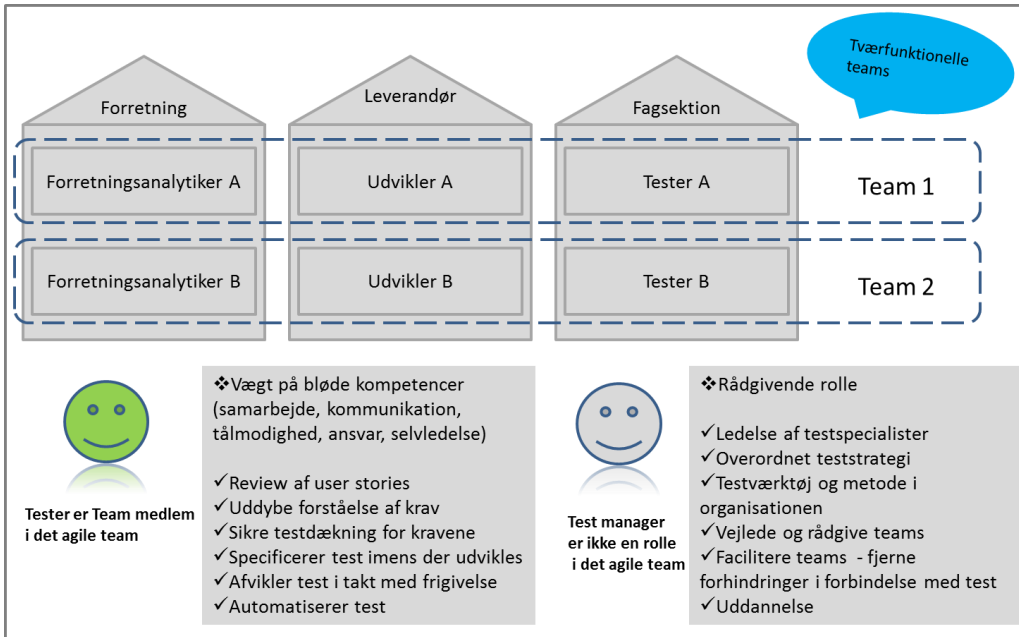
2.1.4.1 Testestens rolle

Det agile team sammensættes af deltagere fra de relevante dele af forretningen, leverandører og fagsektioner.

Testere fra fagsektioner gennemfører testen i samarbejde med forretningen og udviklere fra en eller flere leverandører.

Test manager fra testelementet og test koordinator fra fagsektioner har en rådgivende rolle ved udarbejdelse af teststrategi og anvendelse af værktøjer og metode.

Især ved større projekter vil der være behov for at udpege en test manager i tilknytning til projektet og det eller de agile teams.



Figur 6 Testers rolle i det agile team

Testeren skal reviewe user stories så det sikres, at acceptkriterier er dækkende og testbare og herudover specificere og gennemføre test af user stories i takt med at de udvikles. Testeren skal således kommunikere på tværs i teamet med både forretningen og med udviklerne. Dette gælder også ved vedligehold hvor det agile team supporterer en løsning i drift og hvor det vil være RFC'er der testes.

2.1.4.2 Testdokumentation

Test skal dokumenteres ved agil test og der udarbejdes teststrategi, master testplan, testplaner, manuelle og automatiserede test cases, testresultater og defekter samt rapportering fra testen. Indsatsen ved testdokumentation er især fokuseret på den kontinuerlige vedligeholdelse af test cases og automatisering af test.

❖ Teststrategi	<ul style="list-style-type: none"> ✓ THB på organisatorisk niveau ✓ Skabeloner ✓ Fælles standarder ✓ Fælles værktøjer og metoder 	<ul style="list-style-type: none"> ✓ Test-infrastruktur og miljøer ✓ Uddannelse og vejledning ✓ Teststrategi og Master testplan for projekt
❖ Test cases	<ul style="list-style-type: none"> ✓ Test case design foregår i sprintet ✓ Test case har sporbarhed til user story og acceptkriterier ✓ Testdesigntechnikker anvendes ✓ Test cases oprettes i testværktøj ✓ Session sheet anvendes ved exploratory test 	<ul style="list-style-type: none"> ✓ Test cases for user stories gennemføres i sprintet ✓ Defekter oprettes og sættes på sprint backlog ✓ Test cases vedligeholdes i sprintet ✓ Test cases til regressions test udvælges og automatiseres
❖ Test dokumentation	<ul style="list-style-type: none"> ✓ Principperne for 'letvægt' ✓ Testplanlægning pr sprint ✓ Detaljer afklares direkte i teamet med forretningen ✓ Teamet definerer testtilgang som opfylder krav til governance ✓ Testestimat indgår i user story estimat 	<ul style="list-style-type: none"> ✓ Statisk test med review af user stories og acceptkriterier samt kodereview ✓ Regressionstest er vedligeholdt ✓ Teststatus og fremdrift dagligt ✓ Defekter er dokumenteret i testværktøj ✓ Testresultat er en del af sprint demo

Figur 7 Testdokumentation ved agil test



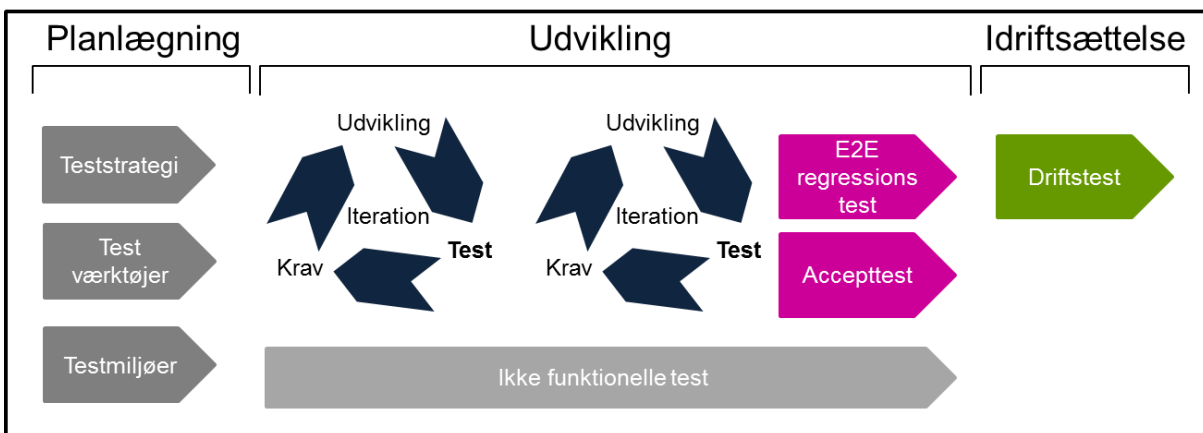
Teststrategi skal udarbejdes som normalt inden projektets start og indeholde en master testplan. Testplan skal udarbejdes ved sprintplanlægningen og kan udbygges i løbet af sprintet.

Test case design, oprettelse af testdata, forberedelse og gennemførelse af test foretages som opgaver i sprintet. Test gennemføres hurtigst muligt i takt med udviklingen af user stories og der kan anvendes explorative test og benyttes 'session sheet' som er en skitse til test cases og testdesigntechnikker for user story og acceptkriterier - der efterfølgende anvendes til det egentlige test case design. Forskellen fra en traditionel tilgang er at testen gennemføres trinvis med udviklingen og testfaser konvergerer sådan der gennemføres alle testtyper fra unittest til accepttest for en user story i sprintet.

Testrapportering foretages dagligt i forbindelse med det daglige sprint møde (f.eks. scrum) og i form af status på test pr user story og defektstatus i sprintet. Testrapportering indgår i rapporteringen af sprint resultat og udvalgte test cases for user stories gennemføres ved sprint demo.

2.1.4.3 Agil test i projekter

Testen skal planlægges overordnet ved udarbejdelse af teststrategi ved projektopstart og herunder opsætning af testmiljøer, testværktøjer og uddannelse. Der anvendes de fælles standarder og foretages en tilpasning til projektets særlige karakteristika og behov.



Figur 8 Test i agile projekter

Særlig ved udviklingsprojekter kan testen af de enkelte user stories tilrettelægges som test i sprint og derudover en tværgående testindsats for tekniske test og brugervenlighed samt en afsluttende accepttestfase inden idriftsættelse.

Ikke funktionelle test

Performancetest gennemføres som et særskilt testforløb på tværs af sprint. Der er ofte behov for særlige kompetencer der trækkes ind i projektet til denne test og et separat performance testmiljø. Centralt er, at der opbygges en performancetest, der kan afvikles med den relevante belastning af data, antal samtidige brugere og transaktioner samtidig med at



systemet monitoreres, der foretages optimering og testen gentages - indtil kravene til svar-tider er opnået..

Konverteringstest kan gennemføres som en separat test eller i det eller de sprints, hvor der udvikles konverteringsprogram – dette afgøres af størrelse og kritikalitet og skal indarbejdes i teststrategien. Testen gennemføres som en eller flere prøvekonverteringer først på et mindre sæt data og indtil prøvekonvertering med det fulde konverterings datasæt.

Brugervenlighedstest kan gennemføres parallelt med udvikling i sprint og f.eks. med anvendelse af mock ups til visualisering inden de endelige skærbilleder og ende til ende (E2E) bruger rejser er færdige. Resultaterne herfra anvendes som input til ændringer af allerede udviklede user stories og detaljering af user stories i back log.

Driftstest

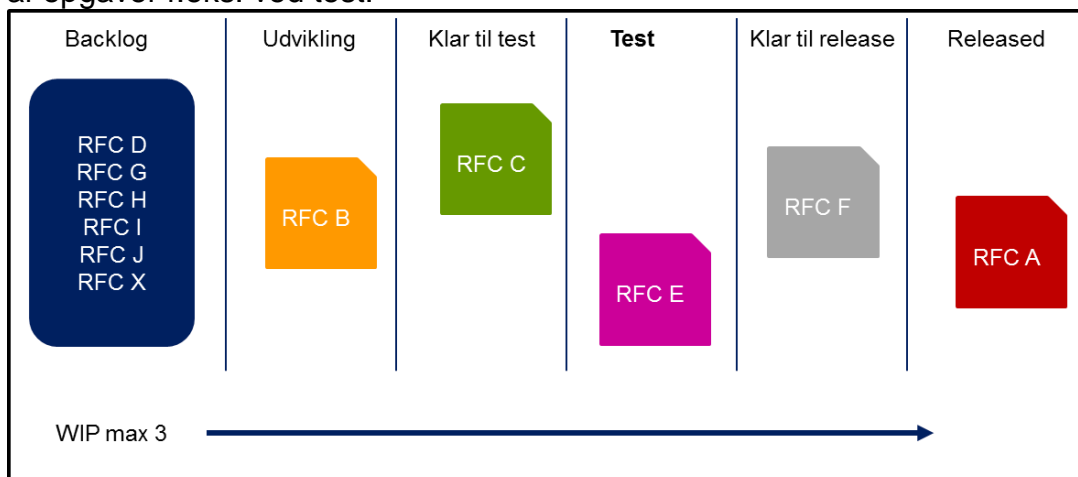
Driftstest gennemføres efter idriftsættelse og ved normal drift og består af målinger af svartider og tilgængelighed samt øvrige driftsprocedurer. Målinger og procedurer etableres og testes inden idriftsættelse. Efter idriftsættelse kan resultater fra driftstest i form af fejl eller forslag til forbedringer fra driften indgå ved vedligeholdelse, så det sikres, at leverancen kontinuerligt lever op til driftskravene. Fejl der først findes i drift f.eks. en batchkørsel der bruger for mange ressourcer, kan være en fejlrettelse/forbedring, der prioriteres til et senere sprint.

2.1.4.4 Agil test ved vedligehold og drift

Ved vedligehold og drift skal der testes inden idriftsættelse af enkelte RFC'er, ved test af opgraderinger mv. og ved regressionstest for en release.

Det agile team kan f.eks. benytte en 'tavle' (gerne digital) hvor de enkelte RFC'er gennemgår processen fra udvikling og test til idriftsættelse. Det agile team er organiseret omkring en specifik løsning og består af både forretning, udviklere og testere som skal beskrive, udvikle og teste RFC'er inden idriftsættelse.

Backlog prioriteres af forretningen og teamet tager selv nye opgaver fra backlog så snart der er ressourcer til det. Afhængig af teamets ressourcer og erfaring besluttet hvor mange opgaver der max må være i en 'fase' ad gangen, dette sikrer at der ikke opstår ophobning af opgaver f.eks. ved test.





Figur 9: Testforløb fra udvikling til test under vedligehold og drift

Opdeling af tavlen kan være anderledes - men det er vigtigt at teamet har besluttet hvornår en opgave er klar til test og hvornår den er klar til release (Definition of done). Leverance til drift kan være flydende, så der idriftsættes kontinuérligt eller organiseres i 'sprint' af f.eks. 2 ugers længde og hvor der idriftsættes efter hvert sprint.

Regressionstest

- Regressionstest før release er en væsentlig testopgave
- Regressionstest bør være automatiseret
- Regressionstest gennemført kan være et kriterie for at den enkelte RFC er klar til release
- Regressionstest som gennemføres kontinuerligt sikrer at fejl findes tidligere
- Defekter fra regressionstest som ikke forhindrer release tilføjes backlog



3. Testprocessen

Testprocessen beskriver de testaktiviteter, som gennemføres ved test. Testprocessen starter med udarbejdelse af teststrategien, som skal beskrive alle testtyper, der gennemføres for den samlede leverance og hvad de enkelte test omfatter for den specifikke leverance.

Teststrategien skal overholde kravene i THB eller angive hvor og hvorfor der fraviges. Ved test i vedligehold og drift skal der udarbejdes en teststrategi for den enkelte løsning eller portefølje der angiver testtilgang og hvilke test der gennemføres.

Test af leverancen består således af en række standard aktiviteter og leverancer, der tilsammen udgør en struktureret proces for enhver testtype der gennemføres.

Der gennemføres følgende standardaktiviteter:

- Teststrategi fastlægges
- For hver testtype i teststrategien gennemføres flg. aktiviteter
 - Testplanlægning
 - Testanalyse
 - Testdesign
 - Testforberedelse
 - Testafvikling
 - Testrapportering

Der leveres som standard følgende leverancer:

- Teststrategi som dækker den samlede leverance
- For hver testtype leveres:
 - Testplan
 - Testkrav
 - Test cases
 - Testdata
 - Defekter
 - Testrapport

Aktiviteter og leverancer er beskrevet nærmere i afsnittene nedenfor.

3.1.1 Testleverancer

Ved gennemførelse af projekter anvendes Den fællesstatslige IT-projektmodel (SIPMO) som projektledelsesmodel og testaktiviteter og -leverancer udføres og leveres ifølge SIPMO faser jf. skema under pkt. 2, testniveauer. Se beskrivelse af modellen her:

<http://fish.msp.forsvaret.fiin.dk/vaerktoejer/projektledelse/SIPMO/Sider/default.aspx>.

Ved gennemførelse af test under vedligehold gennemføres testaktiviteter og –leverancer jf. Change Management processens faser.



Følgende leverancer fra test skal leveres ved hhv. et projekt og en vedligeholdelsessag:

Leverance	Projekt	Vedligehold	Skabelon*/Værktøj	Beskrivelse
Teststrategi	Ja	Valgfrit	MS Word skabelon – gemmes sammen med løsningsbeskrivelse	Teststrategi udarbejdes samtidig med udarbejdelse af løsningsbeskrivelse og for den samlede leverance
Testplan pr test	Ja	Ja Vedhæftet dokument er valgfrit	MS Excel skabelon - testplan uploades til ALM QC Test Plan mappen og faneblad Attachments	Testplan på operationelt niveau
Testdata pr test/miljø	Ja	Ja	Beskrives i testplan og/eller indarbejdes i test cases	Specificerer hvilke data der anvendes, fortrinsvist ved særlige databehov
Testkrav pr test	Ja	Ja	ALM QC	Kravene fra kravspecifikationen. Krav indsættes og mappes til test cases.
Test cases pr test	Ja	Ja	ALM QC	Test cases oprettes sådan krav og løsning dækkes fyldestgørende af testen
Test set pr test	Ja	Ja	ALM QC	Test cases oprettes i 'test set' i QC hvorfra testen kan afvikles
Defekter pr test	Ja	Ja	ALM QC	Defekter oprettes fra testafviklingen ved afvigelser imellem forventet og aktuelt resultat
Testrapport pr test	Ja	Valgfrit	MS Word skabelon - uploades til ALM Test Plan mappen og faneblad Attachments	Status pr test ved testens afslutning
Teststatus	Valgfrit	Valgfrit	MS Excel skabelon	Ad hoc teststatus pr testniveau

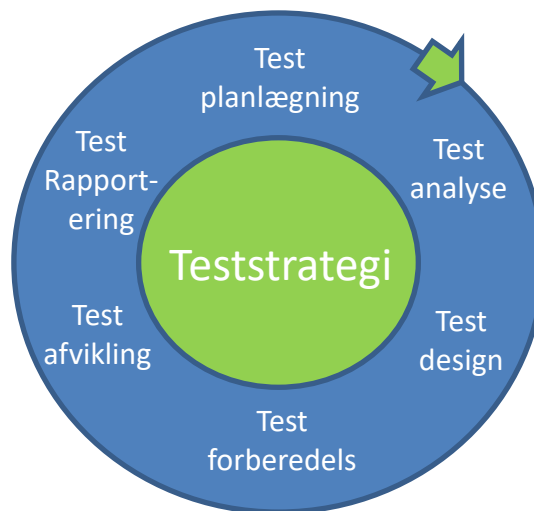
Skabeloner findes som bilag til THB. I det omfang der er udarbejdet en skabelon, skal denne anvendes.

Skabeloner findes under Håndbøger/Testhåndbog her:

<http://fish.msp.forsvaret.fiin.dk/vaerktoejer/projektledelse/Projektskabeloner/Sider/H%C3%A5ndb%C3%B8ger.aspx>



3.1.2 Testaktiviteter



Figur 10: Testproces og testaktiviteter

Teststrategi

Teststrategi udarbejdes med udgangspunkt i THB krav til test, projekt- eller release plan, kravspecifikation og løsningsbeskrivelse.

Teststrategien indeholder:

- Definition af testforløbet: Hvilke test gennemføres, på hvilke miljøer og i hvilken konfiguration.
- Definition af testscope: For hver test beskrives **hvad** der testes (krav, funktionalitet, interfaces, roller mv.). Herudover beskrives krav til testdata og testaktiviteter og ansvarsfordeling.
- Aftaler om testorganisation og risikostyring samt eventuelt særlige forudsætninger for testen.
- Teststrategi godkendes af ITA, LEV, Test Manager og Release Manager.

Testplanlægning

Der udarbejdes en testplan for hver enkelt testtype baseret på teststrategien og de aftalte ressourcer til test.

Testplanen beskriver:

- **Hvordan testen planlægges gennemført.** Der skal beskrives fremgangsmåde/tilgang for testen, tildelt testmiljø og behov for udstyr, faciliteter og testdata.
- **hvem/hvornår** dvs. navngivne roller og ansvar samt tid.

Dette detaljeres yderligere efter design, hvor der indsættes en testafviklingsplan og testindsatsen estimeres med hvem der tester hvad hvornår (kan udelades, hvis det vurderes tilstrækkeligt med testplanen i ALM/QC).

Testanalyse

Baseret på testscope beskrevet i teststrategien og kravspecifikationen anvendes analysefasen, til at formulere de krav, som testen skal verificere og validere.



Kravene dokumenteres pr test i ALM/QC under Requirements modulet. Der oprettes både funktionelle og ikke-funktionelle krav og det sikres, at kravene er testbare og at der er sporbarhed til kravspecifikationen.

Testdesign

I testdesign opbygges testplan i ALM/QC. Der identificeres en eller flere test cases pr. krav og alle relevante test cases oprettes i ALM/QC under 'Test Plan' modulet, hvor de enkelte test cases linkes op mod tilhørende krav.

Test cases beskrives trin for trin. Hvert trin i test casen indeholder instruks om, hvordan brugeren skal agere, samt hvilke data der skal anvendes i testen. Hvert trin i en test case indeholder også det forventede resultat.

Testdata som anvendes i testen, kan være dokumenteret i særskilt ark, sådan det i højere grad er synligt, hvilke testdata der anvendes, f.eks. stamdata med særlige karakteristika.

Testforberedelse

Forberedelsesfasen foretages delvist parallelt med design.

Der udarbejdes testdata på baggrund af de behov for test data, som er identificeret i de enkelte test cases samt tværgående.

Herudover vurderes, om der er behov for særlige data eller større mængder af data evt. data fra andre systemer, der skal loades som en del af testen.

Der bestilles oprettelse af testbrugere.

I forberedelsesfasen efter design er færdigt samles alle test cases i test sets. Et test set består af en række test cases, der logisk hører sammen og skal gennemføres samlet f.eks. i en bestemt sekvens. Dette finder sted i ALM/QC Test Lab modul, hvor test cases samles i såkaldte test-sets.

Testafvikling

Testafvikling omfatter tre dele: Afvikling af testen, rapportering af fejl, gentest efter fejl er rettet.

Samtlige test afvikles i ALM/QC Test Lab modulet og testresultat registreres. Afvigelser imellem forventet resultat og faktisk resultat oprettes som defekter ('defects') i ALM/QC i relation til den fejlede test (se afsnit 5.1 nedenfor).

Det beskrives i defekten, hvad der fejler og fejlen dokumenteres f.eks. ved at vedhæfte kopi af skærbillede, hvor fejlen opstod.

Testrapportering

Under hele testforløbet bliver fremdrift og testens status monitoreret og rapporteret på statusmøder. Ved testens afslutning udarbejdes en testrapport over testforløbet.



Der rapporteres løbende og ved testens afslutning på den planlagte test og med udtræk fra ALM/QC på kravstatus, testafviklingsstatus samt defektstatus.

3.1.3 Vedligeholdelse af test og regressionstest

Regressionstesten skal være vedligeholdt, sådan at test cases løbende opdateres i takt med ændringer i systemet. Herunder skal der løbende tilføjes nye test cases, hvor der er udvidelser til systemet.

3.1.4 Roller og ansvar i forbindelse med test

I tabellen nedenfor er indsat roller og ansvar for de testaktiviteter, der skal udføres. I testplanen skal det fremgå, hvem der har hvilke roller og ansvaret skal uddybes med de konkrete opgaver.

Rolle	Ansvar
Testkoordinator, koncernen	Testaktiviteter og leverancer for FSV test: <ul style="list-style-type: none">• Teststrategi• Testplanlægning• Testanalyse• Testrapportering fra FSV test• Visitering af defekter under FSV testafvikling Øvrigt: Koordinering og ledelse af testaktiviteter
Testkoordinator LEV	Testaktiviteter og leverancer for LEV test: <ul style="list-style-type: none">• Teststrategi• Testplanlægning• Testanalyse• Testrapportering fra LEV test Visitering af defekter under LEV testafvikling Visitering af defekter fra FSV testafvikling
Tester ITA	Testaktiviteter for ITA test: <ul style="list-style-type: none">• Testforberedelse• Testdesign• Testafvikling
Tester FA	Testaktiviteter for FA test: <ul style="list-style-type: none">• Testforberedelse• Testdesign• Testafvikling
Testelementet	Uddannelse i testværktøj, oprettelse af projekt i ALM/QC samt oprettelse af brugere, rådgivning om test.

I tabellen nedenfor er indsat de roller, som reviewer og godkender testleverancerne.

Review og godkendelse	Rolle
Review og godkendelse af teststrategi	ITA, LEV, Test manager, Release manager
Review og godkendelse af testplan for LEV test	Testkoordinator ITA
Review og godkendelse af testrapporter fra LEV test	Testkoordinator ITA, Test manager
Review og godkendelse af testplan for koncernens test	Projektleder/ITA teamleder eller systemejer
Review og godkendelse af testrapporter fra kon-	Projektleder/ITA teamleder eller systemejer



cernens test	Styregruppeformand eller Vedligeholdelsesansvarlig
--------------	--

3.1.5 Start- og slutkriterier for test

De overordnede start- og slutkriterier er defineret nedenfor. Der kan for den enkelte leverance være afvigelser herfra, som dokumenteres i teststrategien.

Vedrørende defektkategorier se afsnit 5 senere i dette dokument.

Unit- og unitintegrationstest	
Startkriterier	<ul style="list-style-type: none">Ingen
Slutkriterier	<ul style="list-style-type: none">Testdækning for systemets komponenter er dokumenteretTestens resultater er dokumenteret i ALM QCIngen udestående defekter i kategori Presserende eller HøjDer kan undtagelsesvis aftales at gå videre med enkelte defekter i kategorierne Medium og Lav, hvis der findes en accepteret handlingsplan for disseTestrapport er udarbejdet og leveret

Systemtest, Systemintegrationstest, Forretningsaccepttest	
Startkriterier	<ul style="list-style-type: none">Testplan er udarbejdetSmoketest er gennemført uden defekter i kategori Presserende eller Høj
Slutkriterier	<ul style="list-style-type: none">Alle planlagte test er gennemførteTestens resultater er dokumenteret i ALM QCIngen udestående defekter i kategori Presserende eller HøjDer findes en accepteret handlingsplan for udestående defekter i kategori Medium eller Lav Der kan undtagelsesvis aftales at gå videre med enkelte defekter i kategorierne Medium og Lav, hvis der findes en accepteret handlingsplan for disseTestrapport er udarbejdet og leveret

Regressionstest	
Startkriterier	<ul style="list-style-type: none">Testrapport for alle forudgående testtyper er leveretTestplan for Regressionstest er udarbejdet
Slutkriterier	<ul style="list-style-type: none">Alle planlagte test er gennemførteTestens resultater er dokumenteret i ALM QCIngen udestående defekter i kategori Presserende eller HøjDer kan undtagelsesvis aftales at gå videre med enkelte defekter i kategorierne Medium og Lav, hvis der findes en accepteret handlingsplan for disseTestrapport er udarbejdet og leveret



4. Testdækning

I dette afsnit beskrives den trinvise test og hvad de enkelte testtyper skal dække mere detaljeret og hvilke egenskaber der skal dækkes af testen.

Det beskrives, hvilke metoder og teknikker der skal anvendes for at identificere test cases og opnå den ønskede testdækning.

Herudover beskrives krav til testdata og principper for anvendelse af testdata.

4.1 Testens scope

I tabellen nedenfor beskrives, hvad der skal testes i den enkelte test for hver testtype. Dette anføres i teststrategien for de enkelte leverancer med det konkrete og relevante indhold.



Testtype	Hvad testes
Installationstest	<p>Det testes at leverancen kan installeres som beskrevet i installationsvejledning. Det verificeres at:</p> <ul style="list-style-type: none">• Leverancen kan installeres jvf instalaltionsvejledning• Fejl logges som beskrevet• Leverancens komponenter er tilgængelige• Leverancens komponenter er integreret som beskrevet• Leverancen er integreret i systemlandskabet• Dataflow i leverancen kan gennemføres ende til ende
Unittest	<p>Det skal testes, at systemet fungerer ifølge designet af de enkelte komponenter, f.eks. programspecifikation, skærbilledesign, rapportdesign, datamodel, rolle-matrix.</p> <p>Der testes de enkelte komponenter i leverancen:</p> <ul style="list-style-type: none">• Skærbilleder• Metoder• Batchkørsler• Workflows• Dataloads• Rapporter• Formularer• Services• Grænseflader• Entiteter• Roller og autorisationer
Unitintegrationstest	<p>Det skal testes, at systemet fungerer ifølge systemets interaktionsdesign, f.eks. tilstandsmodel, workflow specifikation, servicemodel, interfacespecifikation, sekvensdiagram,</p> <p>Der testes integration imellem de enkelte komponenter i leverancen. F.eks.:</p> <ul style="list-style-type: none">• At der teknisk er etableret integration imellem systemets komponenter• at data der overføres til datawarehouse kan anvendes i rapportering• at data der arkiveres, kan fremsøges i arkivet,• at trin i et workflow kan afvikles i den specificerede sekvens,• at en sag kan gå fra status åben til lukket ved specifik hændelse• At en funktion kan kaldes fra brugergrænsefladen.
Systemintegrationstest	<p>Det skal testes at systemet fungerer ifølge systemets grænsefladespecifikation (til andre/eksterne systemer) samt specifikation af de forretningsscenarier, hvor integration til andre systemer indgår.</p> <p>Testen skal omfatte alle de grænseflader til andre systemer, som indgår i leverancen.</p> <p>Der testes trinvist 'indefra og ud':</p> <ul style="list-style-type: none">• at der er 'hul igennem' til det andet system• at grænsefladen kan udfyldes fra systemet som specificeret og modtages i det andet system (eller omvendt)• at fejlsituationer håndteres som specificeret• at data modtaget fra andet system kan indgå i den tiltænkte anvendelse• at scenarier hvor integrationen indgår kan gennemføres 'ende til ende'



Smoketest	<p>Der skal gennemføres en smoketest efter systemet er leveret til testmiljøet. Smoketesten skal verificere, at systemets komponenter er til stede og overordnet fungerer og er klar til systemtest.</p> <p>Testen skal omfatte en regressionstest af den mest centrale funktionalitet i systemet. F.eks.:</p> <ul style="list-style-type: none">• At brugerne kan logge på systemet med de relevante roller• At der kan oprettes, rettes, søges og listes i systemets skærbilleder• At et udvalg af den mest centrale funktionalitet kan gennemføres• At det er muligt at navigere rundt i de forskellige dele af systemet <p>Testen skal udbygges ved hver ny leverance til test.</p>
Systemtest	<p>Det skal testes, at systemet fungerer ifølge systemets funktionelle specifikation. Forretningsprocesdiagram, Use Case diagram, Funktionel specifikation, Rapport-specifikation,</p> <p>Testen skal omfatte systemets funktionalitet:</p> <ul style="list-style-type: none">• Forretningsprocesser• Use Cases• Funktioner• Rapporter• Workflow <p>Testen gennemføres med de relevante roller</p>
Konverteringstest	<p>Det skal testes, at datakonvertering fungerer som specificeret ifølge specifikation af datamapping, konverteringsregler og -procedure.</p> <p>Datakonvertering testes trinvist først med et mindre sæt konstruerede data og herefter med data modtaget fra kildesystemet.</p> <p>Procedure for leverance af data fra kildesystem og load af data i modtagersystem testes.</p> <p>Der gennemføres test i form af en eller flere prøvekonverteringer.</p> <p>Testen skal dække</p> <ul style="list-style-type: none">• at data mappes til de korrekte entiteter og felter• At datatyper konverteres korrekt• At data ikke mistes eller forvanskes under konvertering• At konverterede data efterfølgende kan anvendes i systemet
Performancetest	<p>Testen skal gennemføres, såfremt der er krav til svartider eller det vurderes, at der er risici omkring systemets performance f.eks i forhold til mange samtidige brugere, tidskritiske kørsler o. lign.</p> <p>Testen skal foretages med produktionslignende load af data, transaktioner og brugere.</p> <p>Testen bør foretages på et produktionslignende miljø.</p> <p>Målepunkter for svartider skal afspejle brugerens oplevelse af svartiderne, sådan at svartider måles ved brugergrænsefladen og spredt på de lokationer, hvor systemet skal anvendes</p> <p>Testen skal omfatte:</p> <ul style="list-style-type: none">• Svartidsmålinger• Monitorering af systemets ressourcer <p>Målinger kan f.eks. foretages ved normal transaktionsload og for spidsbelastningssituationer og henover tid.</p> <p>Testforløb: 1) Gennemfør test og monitorér svartid og ressourceforbrug, 2) Analyser resultat. 3) Gennemfør optimering, 4) Gentag test.</p>
Sikkerhedstest	<p>Testens skal gennemføres, såfremt der er specifikke krav til sikkerhed eller hvis der er identificeret sikkerheds risici i forbindelse med leverancen.</p> <p>Testen bør foretages på produktionslignende miljø, dog kan test af adgangskontrol o.lign. foretages som en del af systemtest i testmiljø.</p> <p>Testen kan omfatte:</p> <ul style="list-style-type: none">• Adgangskontrol



	<ul style="list-style-type: none">• Brugeradministration og procedurer• Logning• Revisionsspor• Kryptering af data• Sikkerhedsprocedurer• Overholdelse af sikkerhedsstandarder• Sårbarhedstest
Regressionstest	<p>Regressionstest foretages på tidligere testet funktionalitet og det testes, at der ikke opstår fejl i funktionalitet på eksisterende systemer som følge af den nye leverance.</p> <p>Det analyseres, hvilke dele af det eksisterende system den nye leverance berører og som derfor skal regressionstestes.</p> <p>Derudover kan der være særlig forretningskritiske områder af den eksisterende løsning, som altid ønskes testet ved hver ny leverance.</p>
Forretningsaccepttest	<p>Test af at kravene er opfyldte ifølge kravspecifikationen, forretningsprocesdiagram.</p> <p>Test af, at kravene er opfyldte og løsningen understøtter brugernes arbejdsgange med fokus på brugssituationen. Der testes med forretningsroller.</p> <p>Testen skal omfatte:</p> <ul style="list-style-type: none">• Kravene• Forretningsprocesser ende til ende• Brugssituationen• Forretningsroller• Test med konverterede data
Driftstest	<p>Test af at systemet opfylder kravene til 'Service Level agreement' (SLA) ifølge driftskontrakten samt test af driftsprocedurer.</p> <p>Testen gennemføres i form af en driftsprøve og kan omfatte:</p> <ul style="list-style-type: none">• Oppetider• Svartider• Ressourceforbrug• Kørselstider• Backup/Recovery• Failover• Load balancing• Disaster/recovery• Incident management <p>Testen gennemføres henover en aftalt tidsperiode. Rapportering fra testen bør tilpasses den ordinære driftsrapportering.</p>

4.1.1 Test af systemets egenskaber

Nedenstående skema viser en generel beskrivelse af egenskaber i forbindelse med test samt hvilke egenskaber, der skal testes. De enkelte egenskaber dækkes i de forskellige testtyper.

Egenskab	Generel beskrivelse	Testes - Ja/Nej
Komplethed	At al funktionalitet er til stede og at der ikke er overflødig funktionalitet.	Ja



Egenskab	Generel beskrivelse	Testes - Ja/Nej
Nøjagtighed	At funktionaliteten fungerer korrekt	Ja
Robusthed	At systemet er robust overfor ugyldige eller uventede data og situationer; herunder sikring af validering af input (Negativ test – fejlmeddelelser og advarsler).	Ja
Ydeevne og belastning	At svartider og kørelstider overholder specifikationer og at systemet reagerer hensigtsmæssigt i spidsbelastningssituationer..	Ja
Sikkerhed	At systemet lever op til datasikkerhed, adgangskontrol, kryptering og forhindring af uheld og ulovligt brug.	Ja
Flytbarhed	At systemet kan flyttes til anden leverandør og/eller platform.	Nej (medmindre der er specifikke krav til dette i kontrakten.)
Installation	At systemet kan installeres på de specificerede maskintyper og operativsystemet.	Ja (iflg. INF procedure)
Forvaltning	At systemets kildekode og dokumentation er hensigtsmæssigt opbygget og dokumenteret, så fejlrettelse, tilpasning og udbygning let kan overdrages til personer uden stor erfaring med systemet.	Ja. Det verificeres, at systemet overholder den anvendte programmeringsstandard.)
Ensartethed	At systemet opleves ensartet af brugeren.	Ja. Kan medtages ved forretningsaccept-test
Brugervenlighed	At systemet opleves let at betjene af brugeren.	Ja. Kan medtages ved forretningsaccept-test
Genbrug	At der i systemet er genbrugt relevante komponenter og at der er bidraget til fremtidig genbrugspulje. At der anvendes de gængse standarder på området, som muliggør genbrug.	Ja. Det verificeres, at systemet overholder Programmeringshåndbogen (PGH)

Test af egenskaber fordelt på testniveauer:

Teststrategien skal pr testniveau beskrive, hvilke egenskaber der inddrages i testen og hvad der testes pr. egenskab. Skemaet nedenfor viser den typiske fordeling af test af egenskaber på testniveauer.

Testniveau \ Egenskab	Unit- og unitintegrationstest	Systemtest- og systemintegrationstest	Forretningsaccepttest	Perfomancetest	Sikkerhedstest	Driftstest
Komplethed	X	X	X			
Nøjagtighed	X	X	X			
Robusthed	X	X	X			
Forvaltning	X					
Genbrug	X					
Installation	X					
Ydeevne og belastning				X		X
Sikkerhed	X	X	X		X	



Testniveau \ Egenskab	Unit- og unitintegrationstest	Systemtest- og system-integrations-test	Forretnings-accepttest	Perfomancetest	Sikkerhedstest	Driftstest
Ensartethed	X	X	X			
Brugervenlighed			X	X		

4.2 Testmetoder og teknikker

Test skal baseres på de aftalte/godkendte specifikationer afhængig af, hvilket testniveau der testes jf. V-modellen for test ved en traditionel testtilgang. Ved agil test gennemføres test i det agile team som beskrevet i afsnittet Agil test.

Testen skal verificere, at systemet fungerer som specificeret, identificere eventuelle fejl og validere at forretningens krav er opfyldte.

Specifikationerne er input til, hvad der skal dækkes i testen f.eks.: teknisk design, komponentmodel, datamodel, sekvensdiagram, rollematrix, servicespecifikation, use case diagram, grænsefladespecifikation, forretningsprocesdiagram, kravspecifikation, user story.

Ifølge V-modellen bør den enkelte test forberedes parallelt med specifikationen, som beskrevet under V-modellen ovenfor.

4.2.1 Sporbarhed til Krav

Testen skal have sporbarhed til kravene i kravspecifikationen. Dette betyder, at testen skal dokumentere, at kravene er dækket.

Kravene bør være gjort testbare og have målbare acceptkriterier tilknyttet.

Krav skal specificeres i ALM/QC 'Requirements' modulet og samtlige krav skal være omfattet af en eller flere testcases.

Det er dokumenteret i systemspecifikationerne, hvordan kravene er løst og implementeret og det skal sikres forud for testen, at kravene er adresseret i løsningen. Dette kan gøres ved review og godkendelse af systemspecifikationen inden udvikling påbegyndes, suppleret med prototyping eller andre teknikker der kan anvendes til at visualisere, hvordan kravene opfyldes i systemet.

Ved anvendelse af agil test skal test cases i stedet have sporbarhed til user stories og acceptkriterier.

4.2.2 Use case baseret test

Den primære testdesignteknik, der anvendes, er baseret på use cases eller user stories, forretningsprocesser, funktioner eller brugerscenarier. Afhængig af leverancens karakter anvendes systemets funktionalitet og forretningsprocesser som den primære testenhed.



Forretningsprocesser som indgår i løsningen, skal være dækket af test cases. Test cases skal dække alle trin i processen og de forskellige hændelser, der kan forekomme i processen – herunder variationer i processen og fejlsituationer.

I ALM/QC 'Test plan' modulet oprettes en folder pr. forretningsproces og herunder opbygges de test cases, som tilsammen dækker forretningsprocessen. Test af samtlige trin i forretningsprocesser giver først mening på systemniveau, hvorimod de enkelte skærmbilleder, regler og valideringer mv. som indgår i forretningsprocessen testes på unit- og unitintegrationsniveau.

4.2.3 Risikobaseret test

Test prioriteres på grundlag af forretningsmæssige risici (hvor har det størst forretningsmæssig konsekvens, hvis der opstår fejl) og produktrisici (hvor er der høj kompleksitet, ny teknologi o.lign.).

Risici beskrives i teststrategien og anvendes ved prioritering af test (hvor bør der være en høj testdækning og hvordan prioriteres afvikling af test). Herudover kan der være andre tiltag i test f.eks. at forretningsens slutbrugere deltager i dele af unittest.

4.2.4 Tidlig test og samarbejde om test

Det er forudsat, at parterne (FA, ITA og LEV) i samarbejde planlægger, gennemfører og rapporterer testen af en leverance efter principperne i THB.

Test skal gennemføres tidligst muligt ifølge den trinvis test, hvor fejl detekteres og rettes på de respektive testniveauer.

Forretningen bør have mulighed for, at validere den løsning tidligst muligt og løbende. Visualisering af, hvordan kravene løses, overfor forretningen bør foretages inden løsningens endelige implementering f.eks. ved 'prototyping', 'mock ups' eller agil leverancemodel.

4.2.5 Statisk test

Statisk test er en evaluering af systemet og dets komponenter uden at systemet kører f.eks. ved kode-review og evaluering af opfyldelse af programmeringsstandarder, som defineret i PGH eller på anden måde. Hvis det er relevant at udføre statiske tests beskrives disse i teststrategien. (Øvrige review af dokumenter, procedurer mv. som indgår i kvalitetssikringen, beskrives i kvalitetsplanen og ikke som en del af testproceduren).

4.3 Testdata

Generelle krav til testdata:

- Teststrategien skal definere hvilke typer af testdata, der skal anvendes, for de planlagte testforløb og hvordan disse fremskaffes.
- Ansvar for fremskaffelse af testdata skal som hovedregel følge ansvaret for testen.
- Anvendelse af testdata skal koordineres med andre samtidige test på samme miljø.
- Specifikation af testdata skal indgå som en del af aktiviteterne testdesign og testforberedelse. Testdesign og testforberedelse kan ikke afsluttes, før testdata er specificeret.



- Testdata kan specificeres i et regneark, der vedlægges som bilag til testplanen, så det fremgår tydeligt, hvilke data der anvendes.
- Testdata, der indtastes som en del af testgennemførelsen, specificeres i den pågældende test case.
- Hvis testdata, der er en forudsætning for gennemførelse af en test case, produceres i en anden test case, skal gennemførelse af den anden test case specificeres som forudsætning for test casen.
- Testdata skal være realistiske og repræsentative. Testdata skal dække relevante variationer både i type og antal.
- Ved konverteringstest skal kildesystem levere testdata til brug for testen. Modtager af data kan stille krav til, hvilke data der ønskes leveret af hensyn til testens dækning.
- Ved systemintegrationstest skal der foretages en koordinering af testdata imellem parterne på hver side af grænsefladen. Data skal være repræsentative for begge sider af grænsefladen samt dækkende for udfyldelse af grænsefladen inklusiv test af fejlsituationer.

Testdatasæt:

- Der kan for nogle systemer leveres testdatasæt fra produktion. Dette datasæt loades som en del af testforberedelsen på det relevante miljø, så testen kan gennemføres med produktionslignende data.
- Data kan/skal herudover anonymiseres for personhenførbare data og andre følsomme oplysninger inden de anvendes til testformål.
- Ved konstruktion af særlige testdata til specifikke testformål og hvor data er en forudsætning for gennemførelse af testen og/eller testens gentagelse, skal det overvejes, hvordan testdata kan rekonstrueres f.eks. via en backup/restore procedure.

Test med konverterede data

- Såfremt der skal foretages datakonvertering som en del af leverancen, skal der gennemføres test med de konverterede data. Dette beskrives i teststrategien.
- Fejl i konverteringsprocedurer skal være fundet og rettet under Konverteringstesten og inden generering af de konverterede data, der skal anvendes i testen.
- Der skal under Konverteringstest gennemføres en eller flere 'test-konverteringer' på konverteringsmiljøet; dette for at afluse f.eks. dataformatproblemer og for at verificere, at det kan lade sig gøre at lade data til testmiljøet på 'rimelig' tid.
- Såfremt de konverterede data stammer fra et produktionssystem, skal det fremgå af teststrategien, om der skal foretages anonymisering, såfremt der indgår personhenførbare data eller andre følsomme oplysninger, der ikke uændret må benyttes til testformål.



5. Testværktøj og procedurer

I KAKI benyttes testværktøjet ALM/QC til dokumentation af testplaner, test cases, testresultater og defekter.

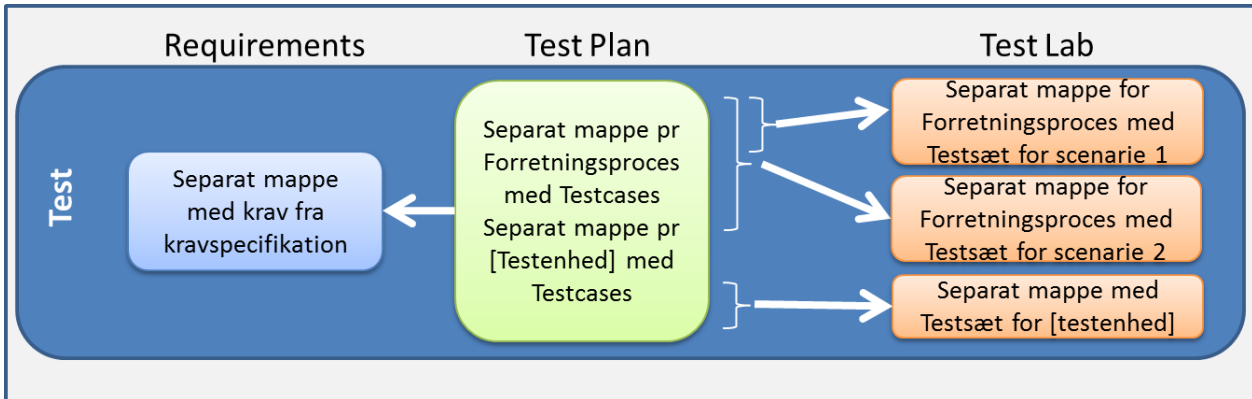
Der er indgået aftale med Forsvarsministeriets Interne Revision om, at dokumentationen skal forefindes her. ALM/QC er således KAKI værktøj til testplanlægning, test gennemførelse og testrapportering.

Gennemførelse af testen kan være implementeret i andre testværktøjer, f.eks. Worksoft, QTP, LoadRunner eller Visual Studio, men test casene skal altid oprettes i ALM/QC og testresultater skal gemmes i ALM/QC, sådan at testen altid styres og monitoreres fra ALM/QC og der herfra kan følges op på gennemførelse og resultat af de planlagte testaktiviteter.

I ALM/QC skal testdokumentation struktureres således:

- Requirement modul
Alle krav til leverancen oprettes med det unikke krav-ID fra kravspecifikationen. Ved agil test oprettes de enkelte user stories når de medtages ved et sprint. Ved vedligehold og drift kan de enkelte RFC'er oprettes men det er ikke et krav.
- Test Plan modul
Der oprettes en folder pr. testenhed jf. hvad der testes - Forretningsproces, Rapport, Roller mv., Testcases indsættes herunder (der kan oprettes underliggende folderstruktur efter behov), Der kan indsættes flere test sæt pr testenhed f.eks. flere scenarier/varianter for samme forretningsproces. Der skal oprettes sporbarhed tilbage til det/de krav, som test case vedrører.
- Test lab modul
Der oprettes den samme folderstruktur som under test plan modulet. Der oprettes test sets hvor test cases fra testplanen samles i de forløb som skal gennemføres. Når disse eksekveres, kan der i fejlsituationer oprettes defekter med direkte reference tilbage test casen. Herved sikres at den relevante test case gentestes ved gentest af fejlrettelser. Yderligere vil der være sporbarhed tilbage til kravene og overblik over, hvilke krav der er påvirket af en fejl.

Strukturen i ALM/QC er illustreret i figuren nedenfor.



Figur 11: ALM struktur

Strukturen skal indeholde sporbarhed til hvad der testes, dette kan som i figuren ovenfor være forretningsprocesser, men kan også være løsningsmoduler, services, integrationer osv.

5.1 Defektkategorier og defektprocedure

Defekter, der findes ved test, skal af testeren klassificeres ved angivelse af fejltipe (se nedenfor). Denne fejltipe kan af testkoordinator og systemejer i samråd justeres under behandlingen.

Defekter fundet i testforløbet kategoriseres som følger:

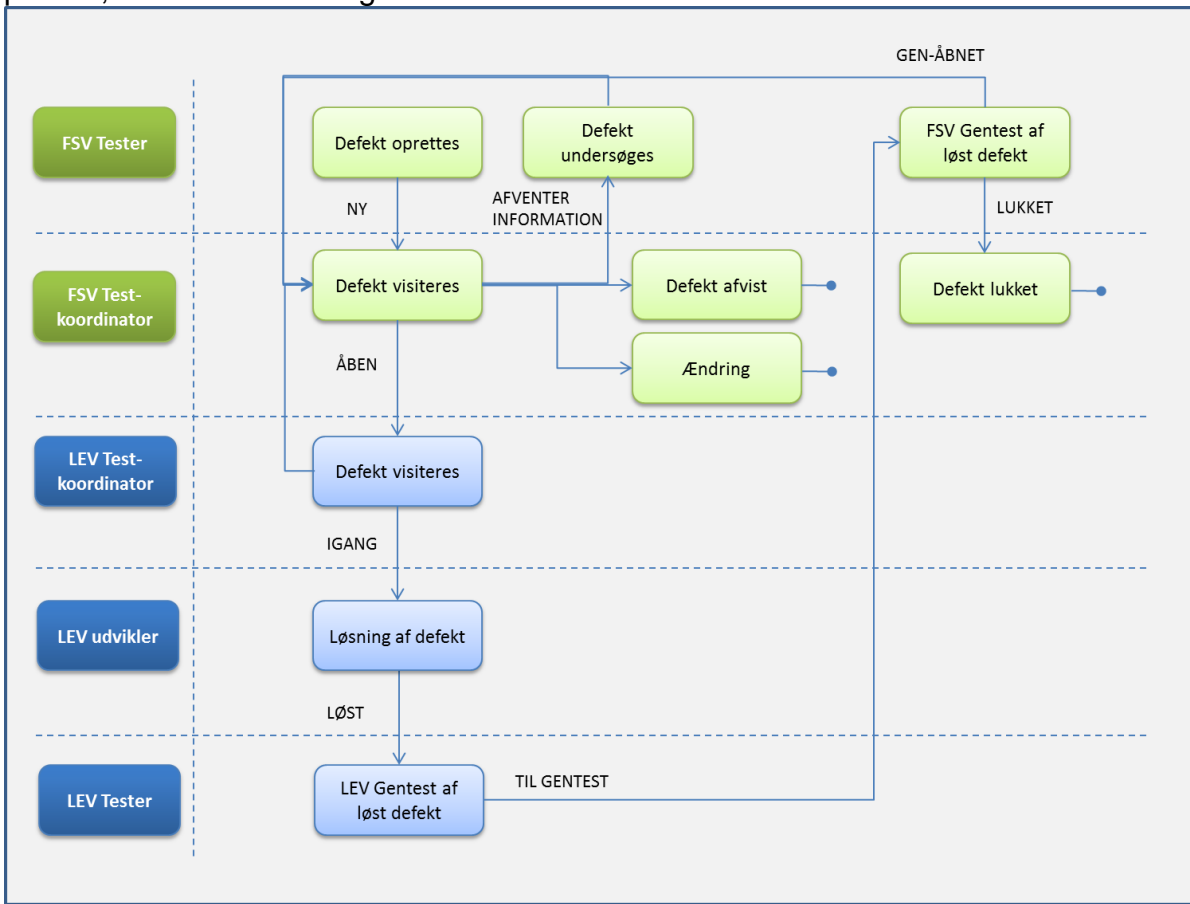
Presserende (Kritisk fejl)	Er en fejl, som medfører, at anvendelsen (testen) af programmet må suspenderes pga. fejlen, eller en fejl, hvor en bruger mister data.
Høj (Betydelig fejl)	Er en fejl, som medfører, at dele eller visse funktioner i programmet ikke kan anvendes eller mangler, og fejlen ikke med rimelighed kan omgås. Dersom en sådan fejl forekommer under et testforløb, skal aktiviteten fortsættes med de kørsler og transaktioner, der ikke er påvirket af den fundne 'Betydelig Fejl'.
Medium (Mindre betydelig fejl)	Er en fejl som ikke forhindrer fuld anvendelse af en funktion (f.eks. testens gennemførelse), eller en fejl som kan omgås.
Lav	Er en fejl i programmet's dokumentation (skriftligt materiale, skærmbillede, hjælpetekst, mv.). Testen skal fortsættes på trods af forekomst af 'Dokumentationsfejl'.

Alle defekter skal rapporteres i ALM/QC i 'Defect' modulet. Defekten skal være relateret til en test case, der er fejlet og skal beskrives, sådan det er muligt at genskabe den.

Vurdering af om testen kan afsluttes med åbne fejl, foretages ud fra de slutkriterier for testen, som er beskrevet i afsnit 3.1.5 ovenfor. Dette betyder, at en test ikke kan afsluttes med åbne fejl i kategorien Presserende eller Høj. Findes der åbne defekter i kategorien Medium eller Lav, kan det undtagelsesvis aftales at gå videre med disse, såfremt der findes en af systemejer accepteret handleplan for fejlrettelse og gentest. Dette skal der redegøres for i testrapporten for den pågældende test.



Defektens livscyklus fra den oprettes til den lukkes skal gennemføres ved en aftalt defektproces, som illustreret i figuren nedenfor.



Figur 12: Defektprocedure

Bemærk at statusnavne er beskrevet generelt i figuren ovenfor og i ALM kan disse opsættes som aftalt i den enkelte sektion eller projekt.

Ved hvert statusskift skal det dokumenteres i defekten, hvad der er foretaget og hvem den videregives (assignes) til.

Når en defekt løses, bør der i defekten være sporbarhed til fejlrettelsen. Dette kan f.eks. være ved angivelse af den patch, transport eller lign. hvormed defekten løses.

Såfremt en defekt i virkeligheden er en ændring til løsningens scope, sættes status på defekten til dette og defekten skal i stedet behandles under procedure for behandling af ændringer..

Såfremt en defekt ikke fører til en fejlrettelse, bør defekt blive afvist som fejl, så gentest og regressionstest kan planlægges for de egentlige fejlrettelser defekten lukkes med kommentar herom, såfremt defekt status 'Afvist' ikke anvendes.



6. Dokumentformalia

6.1 Dokumentplacering

Dette dokument er placeret under:

<http://fish.msp.forsvaret.fiin.dk/vaerktoejer/projektledelse/Projektskabeloner/Sider/H%C3%A5ndb%C3%B8ger.aspx>

6.2 Procedure for opdatering

KAKI ITU er ansvarlig for opdatering af denne standard. Standarden opdateres efter behov.

Udarbejdelse og godkendelse af retningslinjer foretages af KAKI Testelement og forretningsområderne DeMars og IT-udvikling (ITU).

6.3 Godkendelse af fagligt indhold

Seneste revision: December 2020, Test tilgang ved traditionelle og ved agile metoder samt diverse opdateringer

Stabsnummer og navn	Titel/Organisation	Godkendt dato
FMI-KI-DCM02 Olsen, Thomas Beck	Test Manager/KAKI Testelement	16-12-2020
FMI-KI-CHDOP Jensen, Oluf Timo Dybdal	Chef for DOP og DCM /FO DeMars	
FMI-KI-CHKOS Jørgensen, Niels Majland	Chef for Styringssektionen og IT-subporteføljekontoret /FO KOA	