

**Overordnet teststrategi
for
Sykehuspartner**

DISTRIBUSJON

| Organisasjon | Navn | Stilling/Rolle |
|----------------|-----------------|----------------|
| Sykehuspartner | Prosessportalen | Alle |

ENDRINGSLOGG

| Versjon | Dato | Kapittel | Endring | Produsent | Godkjent av |
|---------|------------|--------------------------|--|---------------------|---------------------|
| 0.1 | 06.08.2013 | Hele dokumentet | Opprettet | Linh Nguyen | Per-Morten Gullberg |
| 0.2 | 27.08.2013 | Noen kapitler | Mindre endringer etter gjennomgang utført av Per-Morten | Linh Nguyen | Per-Morten Gullberg |
| 0.9 | 20.09.2013 | Generelt hele dokumentet | Mindre endringer og omskrivninger etter gjennomgang utført av Per-Morten | Per-Morten Gullberg | Per-Morten Gullberg |
| 1.0 | 09.10.2013 | Generelt hele dokumentet | Mindre endringer og omskrivninger etter gjennomgang utført av Per-Morten | Per-Morten Gullberg | IKT-direktør |

Innholdsfortegnelse:

| | | |
|----------|---|------------------|
| 1 | <i>Innledning.....</i> | <i>1</i> |
| 1.1 | Målgruppe for dette dokumentet | 1 |
| 1.2 | Mål | 1 |
| 2 | <i>Sykehuspartners Testprosess Modell</i> | <i>2</i> |
| 2.1 | Sykehuspartners Testpolicy og Teststrategi | 2 |
| 2.1.1 | Utarbeide organisasjonens testrammeverk | 3 |
| 2.1.2 | Oppfølging og styring av testrammeverket | 3 |
| 2.1.3 | Revidere testrammeverket | 3 |
| 2.2 | Test management..... | 3 |
| 2.2.1 | Planlegging..... | 3 |
| 2.2.2 | Oppfølging og styring..... | 4 |
| 2.2.3 | Avslutning | 5 |
| 2.3 | Statisk test – QA gjennomgang | 6 |
| 2.4 | Dynamisk test..... | 6 |
| 2.4.1 | Design og Konstruksjon | 6 |
| 2.4.2 | Gjennomføring | 7 |
| 3 | <i>PPM basert Iterativ Utviklingsmodell og Testprosess Modell.....</i> | <i>8</i> |
| 3.1 | PPMs prosjektfaser med testaktiviteter | 9 |
| 3.2 | Ansvarsfordeling for testnivåer | 9 |
| 3.3 | Testnivå | 10 |
| 3.3.1 | Enhetstest..... | 11 |
| 3.3.2 | Integrasjonstest | 11 |
| 3.3.3 | Teknisk installasjonstest | 12 |
| 3.3.4 | Systemtest..... | 12 |
| 3.3.5 | Systemintegrasjonstest | 13 |
| 3.3.6 | Akseptansetest | 13 |
| 3.3.6.1 | Factory Acceptance Test - FAT | 13 |
| 3.3.6.2 | Site Acceptance Test - SAT | 14 |
| 3.3.6.3 | Operational Acceptance Test - OAT | 15 |
| 3.3.7 | Produksjonstest..... | 15 |
| 4 | <i>Retest og Regresjonstest.....</i> | <i>16</i> |
| 5 | <i>Risikobasert testing</i> | <i>17</i> |

| | | |
|-----------|--|-----------|
| 6 | <i>Testomfang</i> | 18 |
| 6.1 | Testobjekter som skal testes | 18 |
| 6.2 | Testobjekter som ikke skal testes | 18 |
| 7 | <i>Testdekning</i> | 18 |
| 8 | <i>Kvalitetssegenskaper som skal testes</i> | 20 |
| 8.1 | Funksjonelle kvalitetssegenskaper | 20 |
| 8.2 | Tekniske kvalitetssegenskaper | 22 |
| 9 | <i>Testtyper og testteknikker</i> | 25 |
| 9.1 | Testtyper | 25 |
| 9.2 | Testteknikker | 25 |
| 10 | <i>Inngangs- og utgangskriterier</i> | 26 |
| 11 | <i>Godkjenningskriterier</i> | 26 |
| 12 | <i>Avbrudds- og gjenopptagelseskriterier</i> | 27 |
| 13 | <i>Testdokumentasjon og rutiner</i> | 28 |
| 14 | <i>Krav og forutsetninger til testmiljø og fysiske omgivelser</i> | 29 |
| 14.1 | Testmiljø..... | 29 |
| 14.2 | Fysiske omgivelser | 29 |
| 15 | <i>Testdata</i> | 29 |
| 15.1 | Sikkerhet ifb. med midlertidige datalagringsenheter..... | 29 |
| 16 | <i>Roller, organisering og ansvar</i> | 30 |
| 16.1 | Bemannings og opplæringsbehov..... | 30 |
| 17 | <i>Avvikshåndtering</i> | 31 |
| 17.1 | Alvorlighetsgrad av avvik | 31 |
| 17.2 | Status avvik i HP ALM | 32 |
| 17.3 | Endringshåndtering | 32 |
| 18 | <i>Testverktøy</i> | 33 |
| 18.1 | Verktøy for administrasjon av testing og tester..... | 33 |
| 18.2 | Verktøy for automatisert funksjonell test | 33 |
| 18.3 | Verktøy for automatisert teknisk test | 33 |
| 18.4 | Testautomatisering | 33 |
| 19 | <i>Generelle prinsipper om testing</i> | 35 |
| | <i>Vedlegg 1 Begreper og forkortelser</i> | 36 |

| | |
|--------------------|----|
| Begreper | 36 |
| Forkortelser | 38 |

1 Innledning

Dette dokumentet representerer overordnet teststrategi for Sykehuspartner (heretter kalt SP).

Det er tatt utgangspunkt i IEEE Std. 829-1998 IEEE Standard for Software Test Documentation, ISO/IEC 29119 Test Process Modell - Software Testing – Part 2 og ISO/IEC 9126-1 Software engineering – Product quality – Part 1: Quality Model.

Strategien er godkjent av Sykehuspartner og skal benyttes i forbindelse med innføring av nye IT-løsninger eller ved endring av eksisterende IT-løsninger. Den gjelder uansett hvilken leveransemodell og utviklingsmetodikk som benyttes, og er egnet for alle prosjektstørrelser. Den gjelder også uavhengig av om det er eksterne leverandører eller om det er en forvaltningsleveranse.

Sykehuspartner v/IKT direktør er ansvarlig for forvaltning av Teststrategien.

Teststrategien skal være et levende dokument som løpende vurderes og tilpasses til de endringer som til enhver tid skjer i Helse Sør-Øst.

1.1 Målgruppe for dette dokumentet

Dokumentet er skrevet for:

- **Sykehuspartner**
 - **Alle som planlegger en testfase** for å gi overordnede retningslinjer og prinsipper for testing
 - **Prosjekteiere, prosjektledere, driftspersonell, delprosjektledere test, testledere** for å gi trygghet for at it-løsningene blir testet forsvarlig, og at det er klart til produksjon når it-løsningene er godkjent
 - **Øvrig personell** som er involvert i prosjekter og leveranser som en overordnet informasjon om hvordan test av it-løsningene skal gjennomføres
- **Eksterne leverandører** til Helse Sør-Øst hvor Sykehuspartner er involvert.
- **Alle kunder** av Sykehuspartner.

1.2 Mål

Testing er en prosess som gir innsikt i, og råd om, kvalitet og tilhørende risiko. Målet for prosessen er å verifisere og rapportere på produktkvaliteten så tidlig som mulig i prosessen ved hjelp av feilstatistikk og risikobasert vurdering, med de ressurser og innen den kalendertid som er til rådighet. Strategien inneholder nødvendige testmetodikker som innebærer praktiske og vesentlige retningslinjer for å bidra til å nå dette målet på en systematisk og forsvarlig måte.

2 Sykehuspartners Testprosess Modell

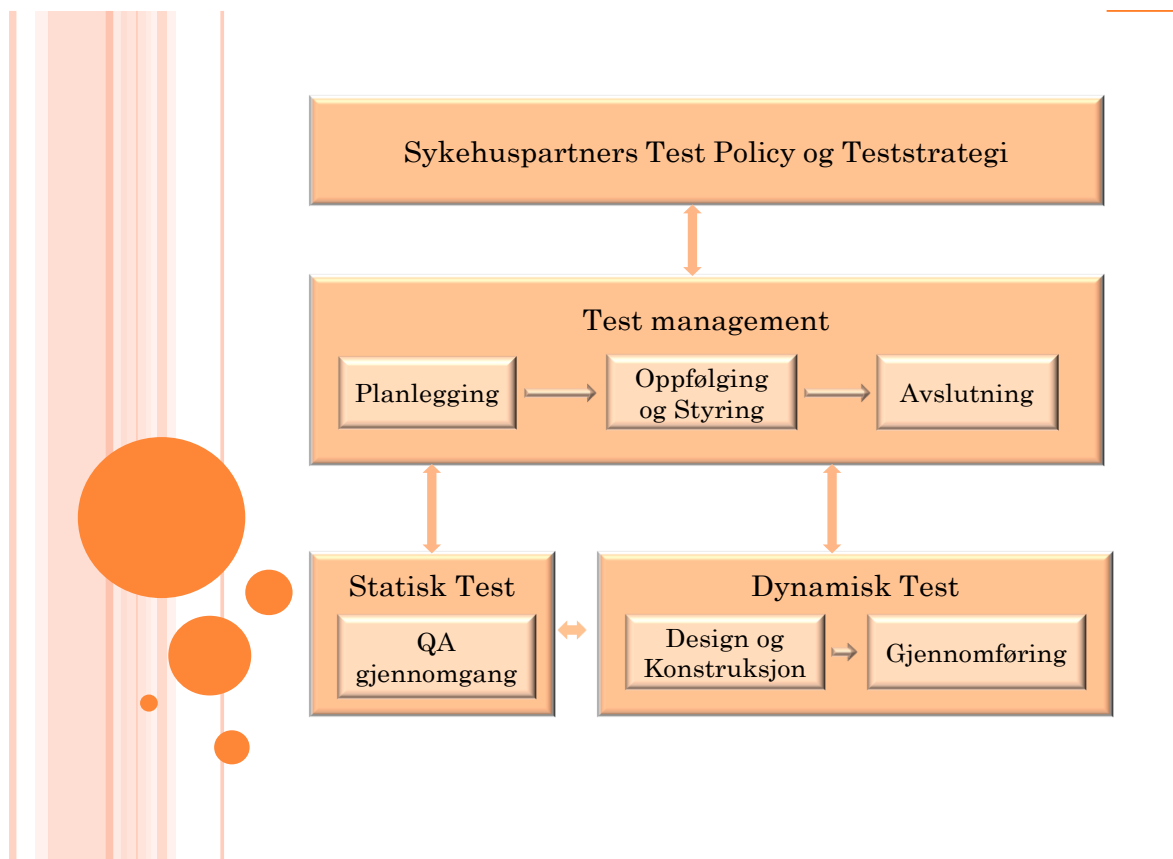
Figuren under viser Testprosessmodellen Sykehuspartner bruker og som er en tilpasning/forenkling av ISO/IEC 29119 Test Process Modell -Software Testing – Part 2.

Modellen består av følgende tre hovedprosesser som er hovedsakelig utført i angitt rekkefølgen.

- Organisasjonens Testpolicy og Teststrategi
- Test management
- Statisk test og Dynamisk test

Hver prosess har en rekke aktiviteter som må gjennomføres og noen av dem kan utføres sine oppgaver på tvert av prosessene.

Følgende underkapitler vil utdype i beskrivelsen av hele Testprosess Modellen.



Figur Testprosess Modell

2.1 Sykehuspartners Testpolicy og Teststrategi

Proessen har følgende aktiviteter som forventes å bli utført i angitt rekkefølge

- Utarbeide organisasjonens testrammeverk
- Oppfølging og styring av testrammeverket
- Revidere testrammeverket

2.1.1 Utarbeide organisasjonens testrammeverk

Aktiviteten består av følgende oppgaver

- Utarbeide testpolicy og teststrategi
- Utarbeide rutiner, maler, sjekklister for ROS-analyse, testplan, bruk av HP-ALM, etc.
- Utarbeide rollebeskrivelser
- Vurdering og implementering av testmetodikk og testverktøy

2.1.2 Oppfølging og styring av testrammeverket

Aktiviteten består av følgende oppgaver

- Sørge for at testrammeverket blir brukt tilstrekkelig og på en forsvarlig måte

2.1.3 Revidere testrammeverket

Aktiviteten består av følgende oppgaver

- Rammeverket skal med jevnlig mellomrom revideres

2.2 Test management

Prosessen har følgende aktiviteter som forventes å bli utført i angitt rekkefølge

- Planlegging
- Oppfølging og styring
- Avslutning

2.2.1 Planlegging

Aktiviteten består av følgende hovedoppgaver

- Utarbeide egen teststrategi eller tilsvarende overordnet/master testplan for testprosjektet. Dette gjelder for større prosjekter. Sykehuspartners felles overordnet teststrategi skal brukes som grunnlag
Teststrategi eller master testplan skal inkludere testprosjektrisiko og overordnet produktrisiko
- Utarbeide testplan både overordnet og detaljert for hvert testnivå
- Planlegge og bestille nødvendige ressurser (kompetanse, tidsbehov, opplæring). Dedikert Defect manager må eventuelt vurderes. Se kap. Roller, organisering av ansvar.
- Organisering av testgruppen og ansvarsfordeling
- Utarbeide budsjett for testprosjekt
- Planlegge hva som skal testes når
- Prioritering av tester basert på risikoanalysen. Se kapittel risikobasert testing
- Sørge for at testmiljøer blir etablert
- Sørge for at testlokaler er tilgjengelig
- Sørge for at testverktøy er tilrettelagt for bruk. HP-ALM evt. tilpasses til prosjektet
- Ordne brukertilgang til systemer, miljøer og testverktøy for alle involverte
- Etc.

Test er en viktig prosess i prosjektløpet og må planlegges godt. Detaljplanlegging skal starte parallelt med aktivitet for design og konstruksjon.

Det skal utarbeides overordnet teststrategi eller overordnet/master testplan for prosjektet. Sykehuspartners overordnet teststrategi brukes som basis. Teststrategien/master testplan skal være et statisk dokument og inneholde avgrensninger og fokus som er spesielle og stabile for hvert prosjekt. Det skal også lages testplan, først overordnet og deretter detaljert.

Normalt vil en testplan dekke alle testnivåene, men for store utviklingsprosjekter kan det være hensiktsmessig å lage en testplan for hvert testnivå. Testansvarlig for det aktuelle testnivået har ansvaret for å fylle ut for sitt testnivå. Testplan skal regelmessig følges opp, oppdateres og justeres. Testplanen må være ferdigstilt og godkjent av prosjektleder før testen starter.

All test som gjennomføres hos SP skal være risikobasert. Et vesentlig mål med planlegging er å optimalisere testarbeidet og fokusere på ”riktige” deler av systemet. Det betyr at de delene av it-løsningen som forårsaker størst skade dersom de feiler i bruk, alltid skal ha høyest testdekning. Tilsvarende gjelder for de deler av it-løsningen der risikoen for feil anses som større enn andre deler.

Det skal gjennomføres en analyse av produktrisiko ved å vurdere følgende:

- Hvilke deler av systemet som anses for å ha størst risiko for feil
- I hvilke deler av systemet vil konsekvensen av feil være størst?
- Kvalitetssegenskapene prioriteres innen hvert funksjonsområde i henhold til pkt 1 og 2

Ut fra disse vurderingene bestemmes det hvilke deler av it-løsningen som bør gis størst prioritet ved testing. Se nærmere på kapittel Risikobasert testing.

2.2.2 Oppfølging og styring

Aktiviteten består av følgende hovedoppgaver

- Ukentlige statusrapporter
- Rapportering (framdrift og kvalitet)
- Prosjektspesifikk rapportering
- Oppfølging av avvik
- Avvikshåndtering
- Overvåke fremdrift
- Forvaltning av problemer, avvik, risiko og hendelser
- Foreta omprioriteringer
- Dokumentere og analysere testresultater

Testoppfølging og styring er observasjon av testaktiviteter og sammenlikning av det som faktisk skjer mot testplanen. I tillegg rapporteres status og avvik fra testplanen, samt at eventuelle tiltak for å nå testmålene iverksettes. Testplanen må kontinuerlig oppdateres som resultat av oppfølgingen.

Testledelse går ikke bare på administrasjon og oppfølging av testprosessen, men også på testmiljø og testverktøy. Status kan mottas både via rapportering fra testere og automatisk generert rapport fra testverktøy. Det avgjøres for hver enkelt leveranse eller innenfor hvert system om det etableres rapportering fra hver enkel tester til testansvarlig.

Gjennomføring av et testnivå starter med et oppstartsmøte ledet av testansvarlig. I dette møtet gjennomgås følgende punkter:

- Oppfyllelse av inngangskriterier
- Detaljert gjennomføringsplan
- Ansvarsfordeling
- Rutiner for melding av observasjoner
- Spesielle forhold ved dette testnivået
- Avvikshåndtering

Under gjennomføringen av testene er Testleder sentral i forhold til oppfølging av planlagte tester. Testere får ansvar for et sett med testobjekter og/eller testprosedyrer og rapporterer daglig til testleder.

Testleder skal sørge for at alle avvik dokumenteres i ALM defect modul. Testleder skal også rapportere minimum ukentlig til prosjektleder. Rapporten vil bestå av en side med kort oversikt over gjennomførte tester, feilsituasjonen og en risikoanalyse.

Observasjonsanalyse er en sentral oppgave i utøvelsen av et testnivå. Testleder er ansvarlig for at dette gjennomføres men det vil i hovedsak være utvikler/arkitekt som utfører feilanalysen i felleskap. Dersom avvikene som er avdekket i testløpet klassifiseres som feil, registreres det en melding for feilretting. Det kan forekomme feil både i systemet og i testopplegget. Begge typene feil skal registreres og rettes. Omfanget av retest og regresjonstest må tydelig avklares. Se kapittel Retest og regresjonstest.

2.2.3 Avslutning

Aktiviteten består av følgende hovedoppgaver

- Testrapport(er) /Test sluttrapport
- Evaluering
- Opprydding av testmiljøer og sensitive data, etc.

Det må sjekkes at testavslutningskriteriene er møtt/oppnådd, og vurderes om testen kan anses som avsluttet eller om det er nødvendig med flere testtilfeller eller gjennomkjøringer.

Testnivået avsluttes med at Testleder utarbeider en testrapport. Testrapporten skal godkjennes av prosjektleder og skal distribueres til prosjektdeltagere, kunde, leverandør og andre interessenter.

Testrapport skal inneholde følgende punkter

- Teststatistikk og testhistorikk
- Dekningsgrad
- Ressurser, kompetanse og samspill mellom interne og eksterne bidragsyttere
- Flaskehalser
- Hva har gått bra
- Forbedringspotensialer
- Utestående feil
- Rettede feil som gjenstår å teste
- Eventuelle endringsønsker som foreligger (til utredning eller beslutning)
- Evaluering av testforløpet opp mot testplanen
- Evaluere hvor og hvorfor feilene oppstår, og eventuelle forbedringstiltak til neste prosjekt (leveranse)
- Anbefaling fra Testleder
- Osv.

I tillegg til testrapporten skal det holdes interne evalueringsmøter, evalueringsmøte med aktuell leverandør samt slutføre alle dokumenter som er benyttet i testprosjektet. Godkjenning, lagring og distribusjon må foregå etter forhånds avtalte prosedyrer.

Til sist i denne aktiviteten skal det ryddes opp i testmiljøet slik at ikke sensitive data blir liggende tilgjengelig.

2.3 Statisk test – QA gjennomgang

Aktiviteten består av følgende hovedoppgave

- QA gjennomgang

Statisk test er en form for QA gjennomgang hvor relevante dokumentasjoner blir gjennomgått for å finne feil.

Å oppdage feil i dokumentasjoner som kravspesifikasjon, løsningsbeskrivelser og testbeskrivelser/testcase i et tidlig stadium i prosjektet er mye billigere enn å gjøre det i et senere stadium av prosjektløpet.

Å foreta en QA gjennomgang betyr at vi validerer/verifiserer eksempelvis at kravspesifikasjon/løsningsbeskrivelse imøtekommer brukernes virkelige behov. QA gjennomgang omfatter også vurderinger av graden av overensstemmelse mellom eksempelvis kravspesifikasjon/løsningsbeskrivelser og testcase/testbeskrivelser eller mellom løsningsbeskrivelser og kravspesifikasjoner.

Revidering av et dokument er også en form for QA gjennomgang. For eks. kravspesifikasjonen eller løsningsbeskrivelsen er ikke oppdatert i forhold dagens leveranse og de må revideres for å kunne bruke som testbasis.

Formålet med QA gjennomgang er å teste/å finne feil i potensielle dokumentasjoner lenge før system/kode er ferdig utviklet. Feil funnet i QA gjennomgang tidlig i prosjektfaser er ofte mye billigere å rette enn de en oppdager når en gjennomfører de normale testnivåene. Hvilken som helst dokumentasjon kan gjennomgås, inklusive kravspesifikasjoner, løsningsbeskrivelser, kode, testplaner, testspesifikasjoner, testcase, testskripter, brukermanualer, etc.

Uformell gransking, strukturert gjennomgang, teknisk gjennomgang og inspeksjon er QA gjennomgang teknikker som er anbefalt å ta i bruk. Disse teknikkene kan gjennomføres internt og eksternt.

2.4 Dynamisk test

Proessen har følgende aktiviteter som forventes å bli utført i angitt rekkefølgen

- Design og Konstruksjon
- Gjennomføring

Forskjellen på statisk test og dynamisk test er at dynamisk test krever kjøring av programvaren for å finne feil, men dette er ikke tilfellet for statisk test.

2.4.1 Design og Konstruksjon

Aktiviteten består av følgende hovedoppgaver

- Utarbeide testene (testcase/testbetingelser) basert på risikoanalysen og testdekningsgrad
- Utarbeide testprosedyrer for testene med utgangspunkt fra testobjekter
- Utarbeid testene (testskript) for automatisert testverktøy
- Lage testdata
- Sørge for at testmiljøet er klart for bruk
- Eventuelt opprette/tilpasse diverse sjekklister, etc.

Testene og testprosedyrer for testobjekter skal beskrives og ferdigstilles i testverktøyet og godkjennes av testansvarlig. Gjenbruk av testprosedyrer vurderes for hvert testnivå. Dersom det er behov for spesielle testdata, skal disse være inkludert i testprosedyrene direkte eller det skal refereres til aktuelt testdataskjema, testdatafil e.l.

I denne aktiviteten skal testbasis samles, organiseres og analyseres. Testbasis er all dokumentasjon som benyttes i spesifisering av testene. Dette kan være kravspesifikasjon, usecase modeller, brukerhistorier, arkitekturskisser, sekvensdiagrammer og lignende. Testere må gjøres kjent med teststrategien, testplanen, testbasis og testobjekter.

Testmiljøet skal nå forberedes og settes opp.

Aktiviteten består også i å vurdere hva som skal testes samt å utarbeide testene på bakgrunn av risikovurdering. Se kapittel for risikobasert testing.

2.4.2 Gjennomføring

Aktiviteten består av følgende hovedoppgaver

- Kjøre testene i HP ALM
- Kjøre testene i automatisert testverktøy
- Registrere avvik i HP ALM
- Analysere avvik

I gjennomføringsaktiviteten anbefales det å kjøre først noen få testprosedyrer for å verifisere at leveransen i det hele tatt virker og at testmiljøet fungerer. Deretter kjøres alle planlagte testprosedyrer manuelt og/eller automatisk. Det er viktig å dokumentere alle testresultatene i HP ALM.

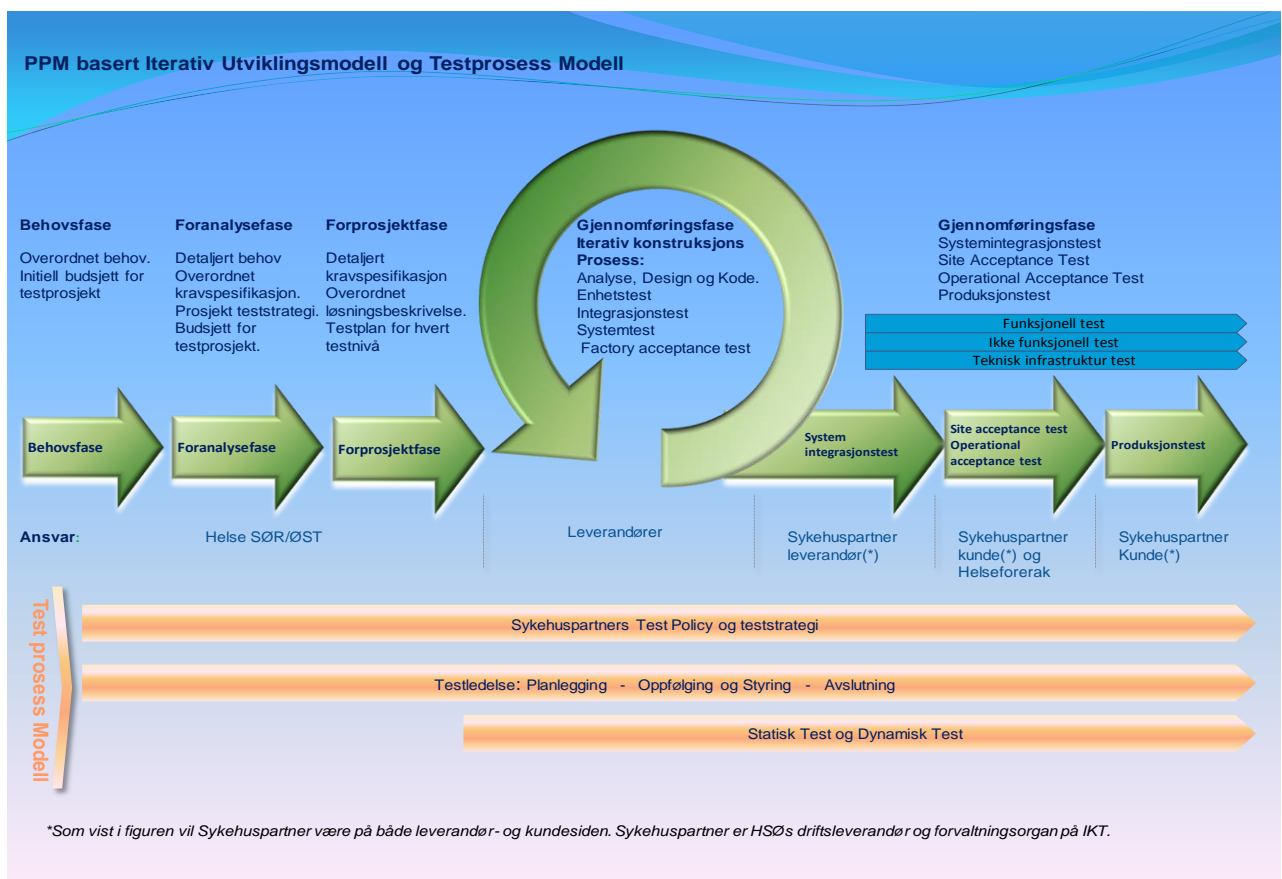
Det må avgjøres om faktisk resultat samsvarer med forventet resultat. Dersom det er avvik, skal dette dokumenteres.

3 PPM basert Iterativ Utviklingsmodell og Testprosess Modell

Figuren under viser PPM - Prosjekt og Porteføljestyling Modellen - basert på Iterativ Utviklingsmodell og Testprosess Modell. Den viser prosjektfaser med sine utviklingsaktiviteter, testaktiviteter og testnivåer. Den viser også testprosess modellen samt beskrivelse av ansvarsfordeling for de enkelte prosjektfasene og testnivåene. Ansvarsfordelingen er basert på dagens SLA - Service Level Agreement - mellom leverandørene og kunden, hvor Sykehuspartner er både som leverandøren og kunden. Vi gjør oppmerksom på at individuelle avtaler som enkelte prosjekter særskilt har inngått med kunden kan ansvarsfordeling bli annerledes.

Teststrategien er bygget opp med generelle faglige testprinsipper og metodikker som kan anvendes i alle typer IT prosjekter.

PPM er en prosjektmodell som benyttes i prosjekter i Sykehuspartner. Ytterligere beskrivelse av dette finnes i følgende underkapitler.



Figur PPM basert Iterativ Utviklingsmodell og Testprosess Modell

3.1 PPMs prosjektfaser med testaktiviteter

Det er essensielt å vite når/hvor i prosjektløpet basert på PPM ulike testaktiviteter må utøves.

Tabellen under gir en oversikt over det.

| Prosjektfase | Testaktivitet | Dokument |
|---------------|------------------------|--|
| Behov | Planlegging | Innledende budsjett for testprosjekt. |
| Foranalyse | Planlegging | Teststrategi (eller overordnet/master testplan) for testprosjekt som inkluderer testprosjektrisiko. Budsjett for testprosjekt. |
| Forprosjekt | Planlegging | Overordnet testplan for hvert testnivå med overordnet produktrisiko analyse (PRA) og testprosjektrisiko analyse. |
| Gjennomføring | Planlegging | Ressursbestillinger Detaljert testplan for hvert testnivå (som identifisert i overordnet testplan) med detaljert PRA og testprosjektrisiko analyse. |
| | Oppfølging og Styring | Avvikshåndtering Ukentlige statusrapporter Rapportering (framdrift og kvalitet) Prosjektspesifikk rapportering |
| | Design og Konstruksjon | Beskrivelse av testmiljø (Bestille) HP ALM – management og krav modulene HP ALM testplan/testcase Testskript i automatisert testverktøy |
| | Gjennomføring | HP ALM Testlab Defectmodul |
| | Avslutning | Testrapport(er) /Test sluttrapport Evaluering |

3.2 Ansvarsfordeling for testnivåer

Under er en beskrivelse av ansvarsfordeling i forhold til testnivåer basert på dagens SLA - Service Level Agreement - mellom leverandør og kunde, hvor Sykehuspartner kan være både leverandør og kunde. Vi gjør oppmerksom på at i individuelle avtaler som enkelte prosjekter har inngått med kunde kan ansvarsfordeling bli annerledes.

Leverandør vil i hovedsak få ansvar for å gjennomføre og dokumentere tester som enhetstest, integrasjonstest, systemtest, factory acceptance test, installasjonstest, systemintegrasjonstest, mens kunden skal gjennomføre site acceptance test, operational acceptance test, produksjonstest samt foreta kontroll under godkjenningsperioden. Se på tabellen under.

| Testnivå | Kort beskrivelse | Ansvarlig |
|-------------------------|--|------------|
| Enhetstest | Enhetstest er en formell test for å avklare om hver enhet (dvs. test av den minste testbare enhet; skjermbilder, batcher, rapporter, brev, komponenter) fungerer uavhengig av andre enheter. | Leverandør |
| Integrasjonstest | Integrasjonstest er en formell test for å verifisere om de enkelte enhetene fungerer sammen med hverandre. | Leverandør |
| Factory Acceptance test | FAT er en test utført av leverandør i leverandørens testmiljø. Målet med testen er å verifisere at systemet fungerer i henhold til spesifikasjonene for å avdekke eventuelle feil før systemet overleveres til installasjon hos kunde. | Leverandør |
| Systemtest | Systemtest består i å teste systemet som helhet, og grunnlaget for testingen er kravspesifikasjonen for systemet. Her tester man om | Leverandør |

| | | |
|-----------------------------|--|--|
| | systemet oppfører seg korrekt i samspill med omgivelsene, og denne testingen blir ofte utført av en egen testgruppe. Systemtest skal ikke bare teste de funksjonelle kravene, men skal også inkludere ikke-funksjonelle tester | |
| Teknisk installasjonstest | Dette er en test som understøtter alle de andre testnivåene. Testen har som mål å fange opp feil i installasjonen av systemet i et produksjonsnært miljø og i integrasjonspunktene til alle systemer det er integrert mot. | Sykehuspartner (leverandør)* |
| Systemintegrasjonstest | System- og systemintegrasjonstest er en formell test for å avklare om et system virker som beskrevet i løsningsbeskrivelser. Den skal også verifisere om løsningen fungerer med hensyn på interne og eksterne grensesnitt. | Sykehuspartner (leverandør)* |
| Site Acceptance Test | Site Acceptance Test - SAT - er en formell test for å avklare om et system virker som beskrevet i kravspesifikasjon. Den skal også verifisere om løsningen fungerer som brukernes behov basert på kravspesifikasjon. Helseforetak har ansvar for funksjonell SAT, men Sykehuspartner for teknisk SAT. | Helseforetak/ Sykehuspartner (kunde)* |
| Operational Acceptance Test | Operational Acceptance Test – OAT - er en formell test for å avklare om systemet er klart for idriftsetting. Den skal verifisere om opplegg eller rutiner rundt installasjon, overvåking, recovery, backup, alerts, etc. lar seg gjennomføre som forventet. | Sykehuspartner (kunde)* |
| Produksjonstest | En verifikasjon/test som utføres i forbindelse med produksjonssettingsdag før sluttbruker tar i bruk løsning. | Sykehuspartner (kunde)* |

*Som vist i tabellen vil Sykehuspartner være på både leverandør- og kundesiden. Sykehuspartner er HSØs driftsleverandør og forvaltningsorgan på IKT.

3.3 Testnivå

Testene deles i følgende nivå:

- Enhetstest
- Integrasjonstest
- Systemtest
- Teknisk installasjonstest
- System
- Systemintegrasjonstest
- Akseptansetest
 - Factory Acceptance Test
 - Site Acceptance Test
 - Operational Acceptance Test
- Produksjonstest

3.3.1 Enhetstest

Består av to hoveddeler, teknisk test og funksjonell test

Teknisk test er inndelt i to undernivåer

- *Test av infrastruktur*
Testen omfatter spesifikke tekniske tester på enkelte programvare og hardware komponenter som ruter, etc.
Denne testen bør helst utføres før oppstart av funksjonell test og ikke funksjonell test.
Se Teknisk infrastruktur teststrategi.
- Ikke funksjonell test
Test av teknisk kvalitetsegenskaper som blant annet pålitelighet, ytelse, feiltoleranse, etc. på enheter.
Se Ikke funksjonell teststrategi.

Funksjonell test

- Funksjonell test av enheter.

Enhetstest er en formell test for å avklare om hver enhet (dvs. test av den minste testbare enhet; skjermbilder, batcher, rapporter, brev, komponenter) fungerer uavhengig av andre enheter. Formålet med enhetstest er å sikre at utviklet kode er i overensstemmelse med detaljert design og oppfyller de kravene som ligger til grunn for detaljert design.

3.3.2 Integrasjonstest

Består av to hoveddeler, teknisk test og funksjonell test

Teknisk test er inndelt i to undernivåer

- *Test av infrastruktur*
Testen omfatter spesifikke tekniske tester på for eks. kommunikasjon mellom to hardware komponenter.
Denne testen bør helst utføres før oppstart av funksjonell test og ikke funksjonell test.
Se Teknisk infrastruktur teststrategi.
- Ikke funksjonell test
Test av teknisk kvalitetsegenskaper som blant annet pålitelighet, ytelse, feiltoleranse, etc. på integrasjoner.
Se Ikke funksjonell teststrategi.

Funksjonell test

- Funksjonell integrasjon mellom enheter.

Integrasjonstest er en formell test for å verifisere om de enkelte enhetene fungerer sammen med hverandre. Risikoen for at det finnes feil i grensesnittet mellom disse enhetene er stor.

Disse testene kan gjøres av spesialister, eller av utviklerne. Integrasjonstest fokuserer ikke på hva enhetene gjør, men på hvordan de kommuniserer med hverandre, som spesifisert i "Overordnet Design". "Overordnet Design" definerer relasjoner mellom enhetene og dette vil si:

- Hva en enhet kan forvente fra en annen enhet i form av tjenester.
- Hvordan disse tjenestene vil bli kalt opp.
- Hvordan de vil bli levert.
- Hvordan håndtere ikke-standardiserte forhold, dvs. feil.

3.3.3 Teknisk installasjonstest

Dette er en test som understøtter alle de andre testnivåene. Utføres vanligvis av testere fra leverandør med bistand fra Sykehuspartner i Sykehuspartner sitt testmiljø. Testen har som mål å fange opp feil i installasjonen av systemet i et produksjonsnært miljø og i integrasjonspunktene til alle systemer det er integrert mot. Dette er systemet SP og HF tester videre i.

3.3.4 Systemtest

Består av to hoveddeler, teknisk test og funksjonell test

Teknisk test er inndelt i to undernivåer

- *Test av infrastruktur*
Testen omfatter spesifikke tekniske tester for å verifisere om infrastruktur-løsningen helhet fungerer tilfredsstillende i forhold til infrastruktur-løsningsbeskrivelsen.
Denne testen bør helst utføres før oppstart av funksjonell test og ikke funksjonell test. Det må ta en vurdering av testomfanget i forhold til hva som er nytt, endret eller ikke blitt testet i tidligere testnivå/testmiljø.
Se Teknisk infrastruktur teststrategi.
- Ikke funksjonell test
Test av teknisk kvalitetsegenskaper som blant annet pålitelighet, ytelse, feiltoleranse, etc. på system.
Testen kan utføres i parallell med funksjonell test, men ofte i eget testmiljø.
Se Ikke funksjonell teststrategi.

Funksjonell test

- Funksjonell test av system.

Systemtest er en formel test og testen foregår normalt i to omganger iht. inkrementell/iterativ utvikling. Løsningen er delt i sprint/deler.

- Når en sprint blir ferdigutviklet, utfører man en systemtest på den.
- Når alle sprint'ene er ferdigutviklet, utfører man en helhetlig systemtest på alle sprint'ene.

Systemtest skal verifisere hvorvidt systemet som helhet fungerer i henhold til spesifiserte krav i løsningsbeskrivelser. Testen vil normal inkludere online hjelp/brukermanualer, brukerstøtte, etc. Omfanget av testen må vurderes i forhold til innholdet i endringen.

Den skal også verifisere om løsningen som helhet fungerer med hensyn på interne og eksterne systemer. På dette testnivået er det ikke fokus på hver enkelt enhet eller hvert enkelt integrasjonspunkt men på hele systemet.

Test av teknisk infrastruktur og ikke-funksjonell test er to vesentlige deler av systemtest. Testen skal gjennomføres i testmiljøet så nært opp til produksjonsmiljøet som mulig.

Regresjonstest og retest er alltid en viktig del av testen. Testen skal være risikobasert. Se også avsnitt for "Kvalitetsegenskaper som skal/skal ikke testes" for å vite hvilke egenskaper som er relevante for testen.

Denne testen utføres ofte av ekstern leverandør i leverandørens testmiljø. Testen har derfor som mål å sjekke om kunden får det som de har bestilt og om kvaliteten på løsningen er god nok til at produktet kan leveres til kunde. Testen har også til hensikt å sjekke at leveransen er komplett (programvare og dokumentasjon).

3.3.5 Systemintegrasjonstest

Består av to hoveddeler, teknisk test og funksjonell test

Teknisk test

- *Test av infrastruktur*
Testen omfatter spesifikke tekniske tester som for eks kommunikasjon mellom en hardware komponent i et kjernenett og en annen hardware komponent i et lokalt nettverk. Denne testen bør helst utføres før oppstart av funksjonell test og ikke funksjonell test. Det må ta en vurdering av testomfanget i forhold til hva som er nytt, endret eller ikke blitt testet i tidligere testnivå/testmiljø.
Se Teknisk infrastruktur teststrategi.

Funksjonell test

- Funksjonell test av system.

Systemintegrasjonstest skal verifisere hvorvidt integrasjoner mellom systemer fungerer i henhold til spesifiserte krav i løsningsbeskrivelsen.

Omfanget av testen må vurderes i forhold til innholdet i endringen. Testen skal ha sterk fokus på verifikasjon av samspill mellom nytt/eksisterende system og andre interne/eksterne systemer, flere plattformer etc. Den skal gjennomføres i testmiljøet så nært opp til produksjonsmiljøet som mulig. Der hvor det finnes avhengigheter mellom systemer, skal disse grundig testes samlet.

Regresjonstest og retest er alltid en viktig del av testen. Testen skal være risikobasert. Se også avsnitt for "Kvalitetssegenskaper som skal/skal ikke testes" for å vite hvilke egenskaper som er relevante for testen.

3.3.6 Akseptansetest

Typiske typer akseptansetester omfatter følgende

- Factory Acceptance Test
- Site Acceptance Test
- Operational Acceptance Test

3.3.6.1 *Factory Acceptance Test - FAT*

FAT er en type akseptansetest som utføres av ekstern leverandør i leverandørens testmiljø, ofte med deltakelse fra kunden. Siden leverandørens testmiljø kan være ulikt eller ufullstendig i forhold til kundens testmiljø kan testomfanget bli begrenset.

FAT er på mange måter forventes å være så likt som mulig med Site Acceptance Test - SAT. Dvs. det som blir testet i SAT må også testes i FAT i den grad det lar seg gjøre. Se beskrivelse av SAT i følgende kapittel.

Testen har som mål å sjekke om kunden får det som de har bestilt og om kvaliteten på løsningen er god nok til at produktet kan leveres til kunde. Testen har også til hensikt å sjekke at leveransen er komplett (programvare og dokumentasjon).

Denne type testen er mest relevant for standard system/applikasjon.

3.3.6.2 Site Acceptance Test - SAT

SAT er en type akseptansetest som utføres i kundens testmiljø.

Testen består av to hoveddeler, teknisk test og funksjonell test

Teknisk test er inndelt i to undernivåer

- *Test av infrastruktur*
Testen omfatter spesifikke tekniske tester for å verifisere om infrastrukturløsningen helhet fungerer tilfredsstillende i forhold til infrastrukturkrav.
Denne testen bør helst utføres før oppstart av funksjonell test og ikke funksjonell test. Det må ta en vurdering av testomfanget i forhold til hva som er nytt, endret eller ikke blitt testet i tidligere testnivå/testmiljø.
Se Teknisk infrastruktur teststrategi.
- Ikke funksjonell test
Test av teknisk kvalitetsegenskaper som blant annet pålitelighet, ytelse, feiltoleranse, etc. på system.
Testen kan utføres i parallell med funksjonell test, men ofte i eget testmiljø.
Se Ikke funksjonell teststrategi.

Funksjonell test

- Funksjonell test av system.

Akseptansetesten sjekker systemet mot ”Kravspesifikasjonen” og ”Behovsanalysen”. Det ligner på systemtesting ved at hele systemet er kontrollert, men forskjellen er:

- System Testing sjekker at systemet ble levert i forhold til Løsningsbeskrivelsen.
- Akseptansetesting sjekker at systemet er levert i forhold til Kravspesifikasjonen, Behovsanalysen og brukernes prosessbeskrivelser.

Akseptansetest er en formell test som skal ha fokus på prosessflyt i løsningen og verifisere at systemet fungerer i henhold til kravene med fokus på brukernes arbeidsprosesser. Testen vil normal inkludere online hjelp/brukermanualer, brukerstøtte, etc.

Det skal også eksempelvis testgjennomføres et fullt batchdøgn (der dette er aktuelt).

Omfanget av testen må vurderes i forhold til innholdet i endringen og testen skal/bør ikke være en kopi av systemtest.

Akseptansetestmiljøet må være mest mulig likt produksjonsmiljø. Test av teknisk infrastruktur og ikke-funksjonell test er også to vesentlige deler av akseptansetest. Her må det sikres god driftbarhet ved å avdekke eventuelle alvorlige feil ved infrastrukturen og driftsopplegget før produksjonssetting.

Kunden/sluttbruker, og ikke utvikler bør alltid gjennomføre akseptansetesten. Kunden/sluttbruker vet hva som kreves av systemet for å oppnå verdi i virksomheten og er den eneste personen som er kvalifisert til å avgjøre dette. Videre skal hele tjenesten levert av SP testes, inkludert brukerstøtte, dokumentasjon, opplæring, etc.

Akseptansetesten skal alltid gjennomføres i tett samarbeide mellom kunden (HF) og SP.

Regresjonstest og retest er alltid en viktig del av testen. Testen skal være risikobasert. Se også avsnitt for ”Kvalitetsegenskaper som skal/skal ikke testes” for å vite hvilke egenskaper som er relevante for testen.

3.3.6.3 Operational Acceptance Test - OAT

OAT er en teknisk test og blir brukt til å etablere en bærekraftig driftsberedskap av et system eller en endring før produksjonssetting. Dette er for å fokusere på å redusere den operasjonelle risikoen.

Det er anbefalt at driftspersonell fra Sykehuspartner deltar aktivt i denne testen.

Testen konsentrerer seg om områder som installasjon, overvåking, alerts, feiltoleranse, gjenoppretting (recovery), integritet, driftsadministrasjon og supportvurdering og sikkerhetskopiering (backup). Testen vil i stor grad basere seg på å utøve alle prosedyrer/rutiner til disse.

Denne testen kan utføres parallelt med FAT, SAT eller som en del av produksjonstest.

3.3.7 Produksjonstest

Består av to hoveddeler, teknisk test og funksjonell test

Teknisk test er inndelt i to undernivåer

- *Test av infrastruktur*
Testen omfatter spesifikke tekniske tester for å verifisere om infrastrukturløsningen helhet fungerer tilfredsstillende i forhold til infrastrukturkrav.
Denne testen bør helst utføres før oppstart av funksjonell test og ikke funksjonell test. Det må ta en vurdering av testomfanget i forhold til hva som er nytt, endret eller ikke blitt testet i tidligere testnivå/testmiljø. Det må også ta hensyn hva som lar seg teste i produksjonsmiljøet. Se Teknisk infrastruktur teststrategi.
- Ikke funksjonell test
Test av teknisk kvalitetsegenskaper som blant annet pålitelighet, ytelse, feiltoleranse, etc. på system. Testomfanget må vurderes i forhold hva som lar seg testes i produksjonsmiljøet. Testen kan utføres i parallell med funksjonell test.
Se Ikke funksjonell teststrategi.

Funksjonell test

- Funksjonell test av system

Produksjonstest er en teknisk og funksjonell verifisering utført i produksjonsmiljøet for å sikre at omlegging til produksjon er skjedd på en korrekt og forsvarlig måte. Testen fokuserer på å verifisere om det nye eller endrede systemet vil fungere i produksjonsmiljøet.

Hovedsakelig betyr dette det tekniske miljøet, og sjekker aspekter som:

- Påvirker det noen andre systemer som kjører i samme driftsmiljø?
- Er det kompatibelt med andre systemer?
- Har det akseptabel ytelse under belastning?
- Fungere alle integrasjoner med andre systemer som det skal?

Disse testene utføres vanligvis av systemforvaltere og/eller annet it-teknisk personale beskrevet i Testplanen.

4 Retest og Regresjonstest

Retest og regresjonstest er aktuelt for alle testnivåer.

Retest

Samtlige feil som er ferdige rettet skal retestes for å verifisere at den spesifikke feilen/endringen er rettet.

Regresjonstest

Risikoanalysen avgjør om hele eller deler av applikasjonen må regresjonstestes.

Regresjonstest er en test hvor man gjentar et utvalg tester for å verifisere at feilrettinger eller endringer ikke har introdusert nye feil eller uønskede effekter, og at systemet eller komponenten fremdeles tilfredsstillende spesifiserte krav. Ved endring/feilretting i en del av systemet kan det ha uforutsette virkninger på andre systemer. På bakgrunn av dette kan det være nødvendig å fastslå at effekten av endringen/feilrettingen ikke har utilsiktede sideeffekter.

Ved enhver endring/feilretting skal behovet for regresjonstesting vurderes ved at alle tester/deler av tidligere tester skal gjentas.

Regresjonstest gjennomføres for å verifisere at en endring ikke har medført følgeføl i andre eller tidligere testede deler av systemet. Denne testen kommer i tillegg til de testene som må kjøres for å verifisere at feilrettingen eller endringen fungerer som spesifisert.

5 Risikobasert testing

Det skal gjennomføres en ROS - Risiko og Sårbarhetsanalyse - av alle testprosjekter, uansett omfang av test. Både testprosjektrisiko og produktrisiko analysen skal utarbeides.

Prosjektrisiko er risiko relatert til ledelse og styring av testprosjekt, eksempelvis mangel på personell, knappe tidsfrister, endrede krav, etc. Produkttrisiko er risiko som er direkte relatert til testobjektet. Dette kapitlet handler imidlertid om den siste nevnte risikotypen.

Målet med risikobasert testing er å konsentrere testingen om de områder/testobjekter som antas å ha størst sannsynlighet for feil, har høyest kostnad ved feilretting og stor ulempe for Sykehuspartners kunder, tap av omdømme eller andre sikkerhetsrisiko som for eks. pasientsikkerhet.

System, programvare- og infrastrukturkomponenter er komplekse. Fullstendig test med alle mulige inputverdier og med alle mulige forutsetninger er umulig. Testingen skal derfor ikke ha et for høyt ambisjonsnivå men heller fokusere testinnsatsen der det er høyest risiko og hvor konsekvensene er store ved eventuelle feil.

Det skal normalt gjennomføres en risikoanalyse av hele leveransen for å identifisere slike områder. Det skal derfor foretas en identifisering av hvilke testobjekter som skal prioriteres i detaljplanleggingen for det enkelte testnivå. Slike testobjekter skal risikovurderes iht. konsekvensverdier og sannsynlighetsverdier. Deretter blir risikoverdien beregnet for hvert enkelt testobjekt ved å multiplisere konsekvensverdier med sannsynlighetsverdier. Risikoverdien representerer hvor høy risiko testobjektet har. Se risikomatrise under.

Risikoanalysen skal gjennomføres i planleggingsfasen av testarbeidet.

| | | | | | |
|-----------------------------------|------------------------------------|---|---|--|--|
| <i>Svært høy</i> 5 | | | | | <i>Høyest risiko = 25</i> |
| <i>Høy</i> 4 | | | | | |
| <i>Middels høy</i> 3 | | | | | |
| <i>Lav</i> 2 | | | | | |
| <i>Svært lav</i> 1 | <i>Lavest risiko = 1</i> | | | | |
| Sannsynlighet / Konsekvens | <i>Minimal konsekvens</i> 1 | <i>Mindre konsekvens for framdrift for kostnad</i> 2 | <i>Merkbar konsekvens for kvalitet for framdrift for kostnad</i> 3 | <i>Stor konsekvens for kvalitet for framdrift for kostnad</i> 4 | <i>Meget stor konsekvens Leveransens målsetning vil ikke oppnås</i> 5 |

Tabell Risikomatrise (Risikoverdi = sannsynlighetsverdi * konsekvensverdi)

Testing på bakgrunn av en risikoanalyse sikrer at det ikke brukes unødvendig mye tid på mindre kritiske endringer. Fokus rettes mot endringer som er kompliserte og mer kritiske. Videre kan vurderingen gi forskjell i testomfang for funksjonelle og ikke-funksjonelle krav.

Ved å kjenne til mulige trusler/konsekvenser på forhånd, kan sannsynligheten for at de inntre reduseres ved at ulike tiltak igangsettes på forhånd eller underveis i testprosessen. I spesielle/alvorlige tilfeller kan det også være hensiktsmessig at deler eller hele organisasjonen er forberedt på eventuelle konsekvenser.

Sannsynligheten for feil vil vanligvis være størst innenfor områdene:

1. Som er nye eller endrede
2. Ny teknologi i Sykehuspartner.
3. Integrasjon av løsninger.
4. Kompleks systemarkitektur og i grensesnittet mellom systemer.
5. Komplisert logikk/kode
6. Løsning med mange tidligere feil.
7. Funksjoner, grensesnitt, datadefinisjoner som er ustabile, dvs. endres hyppig.
8. Løsning produsert under tidspress.
9. Løsning der mange utviklere har vært involvert, inklusive utskiftning av personell.
10. Nye funksjonelle kompetanseområder ved overføring av prosjekter til linjen. Lav funksjonell og teknisk kompetanse øker risikoen.
11. Enheter som oppdaterer data.
12. Programvare som krever stor grad av manuelle rutiner rundt driften.

Ved vurdering av sannsynlighet for feil på overnevnte punkt som for eks. 3, 4, 5 og 9 bør det foretas i samråd med leverandører eller utviklere. De kan gi bl a en prioriteringsliste over kodekompleksitetsgrader på ulike testområder/testobjekter. Jo høyere kodekompleksitetsgrad testobjektet har, desto høyere sannsynlighet for feil blir det.

Det henvises til en egen prosessbeskrivelse for gjennomføring av ROS.

6 Testomfang

Risikoanalysen avgjør testomfanget, dvs. Hvilke

- Testobjekter/deler av systemet det er viktig å teste
- Testobjekter av systemet det er viktig å utføre regresjonstest

6.1 Testobjekter som skal testes

Hvilke testobjekter som skal testes beskrives i testplanen for det enkelte testnivå. I prinsippet skal alle nye og endrede deler av systemet testes.

6.2 Testobjekter som ikke skal testes

Hvilke testobjekter som ikke er gjenstand for test beskrives i testplanen for det enkelte testnivå. Unntak skal begrunnes i testplanen.

7 Testdekning

Testdekning er et mål på i hvor stor grad de ulike scenarioene eller mulige inputverdiene av en funksjon i et testobjekt som faktisk er blitt testet.

Risikoanalysen er med på å definere nødvendig testdekningsgrad, hvilke testteknikker og kvalitetsegenskaper som skal benyttes i hvilke testobjekter.

I enhetstest og integrasjonstest er testdekning et mål på i hvor stor grad all kode/oppsett faktisk er blitt utført under testene eller sagt på en annen måte hvor stor prosentandel av statements/stier/code branches i koden som blir testet.

En komponent eller et testobjekt med høy risiko bør være gjenstand for høy testdekning. I akseptansetest, systemtest eller systemintegrasjonstest vil det for et slikt testobjekt bli en viktig gjenstand for QA gjennomgang av sine relaterte dokumentasjoner tidlig i prosjektet og testen skal også ha høy prosentandel av mulige scenarioer og inputverdier, samt bruk av flere testteknikker. I enhetstest og integrasjonstest vil det for en slik komponent gjennomføres kodegranskning og testen skal dekke flest mulig statements/stier i kode.

Ulike kvalitetsegenskaper og testteknikker vil også påvirke hvor stor testdekning man oppnår.

I testplanen og testrapporten skal det spesifisere krav til testdekningsgrad som må oppfylles på de testobjektene som er ferdig risikovurdert.

Testerne skal starte arbeidet med den testen som vurderes som mest kritisk (har størst risiko) og fortsette med den nest mest kritiske etc. Dette sikrer at de områdene som det er viktigst å teste grundig får en bred testdekning og blir testet tidlig i testperioden.

Hvis testen avsluttes uten at alle planlagte tester er gjennomført må testansvarlig utrede hvilke risiko som er knyttet til de gjenværende testene. Denne beskrivelsen vil inngå i en risikovurdering av de gjenværende testene i forkant av en eventuell produksjonssetting. Beslutningstaker kan avgjøre om dette er en akseptabel risiko ved produksjonssetting, eller om ytterligere tid må settes av til testarbeidet.

Prosjekter som er teknisk kompliserte, har et uklart kravbilde, synlighet i samfunnet, eller har en generelt høy risiko, må det vurderes om det er behov for testdekning utover minimumskravet.

8 Kvalitetssegenskaper som skal testes

Dette kapitlet viser hvilke egenskaper som skal testes/verifiseres i det enkelte testnivå og er ihht. ISO/IEC 9126-1 Software engineering – Product quality – Part 1: Quality Model.

Flere egenskaper må testes gjennom flere testnivåer. Det er viktig å merke seg at det i de detaljerte testplanene skal beskrives og begrunnes hvis det er avvik fra denne tabellen. Hvis det ikke er avvik, henvises det til tabellen i dette dokumentet, ved avvik gjengis de radene i tabeller der det er avvik

8.1 Funksjonelle kvalitetssegenskaper

Følgende egenskaper kan i stor grad testes i ulike testnivåer som en del av funksjonell test og utføres av sluttbrukere, fagpersoner eller forvaltere.

Tabellen under lister opp funksjonelle egenskaper som skal testes på de forskjellige testnivåene.

| Egenskaper ¹ | ET | IT | ST | SIT | FAT/ SAT | PT |
|--|----|----|----|-----|-------------|----|
| Funksjonalitet | | | | | | |
| <i>Velegnethet/Suitability</i> Funksjonalitet fungerer iht. spesifiserte arbeidsprosesser. | | | X | X | X | X |
| <i>Korrekthet</i> Funksjonalitet gir riktige eller forventet resultat iht. spesifiserte brukerens krav. | X | X | X | X | X | X |
| <i>Samspill mellom programmoduler</i> | | X | | | | |
| <i>Samspill med eksterne systemer</i> | | X | X | X | X | X |
| <i>Samspill med annen programvare</i> | | X | X | X | X | X |
| <i>Sikkerhet (mot uønsket adgang)</i> | X | X | X | X | X | X |
| Brukervennlighet og brukbarhet | | | | | | |
| <i>Forståelig</i> | | | X | X | (X) | |
| <i>Enkel å lære</i> | | | X | X | (X) | |
| <i>Effektiv i bruk</i> | | | X | X | (X) | |
| <i>Attraktivitet</i> | | | X | X | (X) | |
| Pålitelighet | | | | | | |
| <i>Funksjonell feiltoleranse og negativ test</i> | X | X | X | X | X | X |
| Samsvar | | | | | | |
| <i>Samsvar med standarder og forskrift</i> | | | X | | X | |

Tabell Oversikt over funksjonelle kvalitetssegenskaper

Funksjonalitet

Velegnethet/Suitability

Funksjonalitet fungerer iht. spesifiserte arbeidsprosesser.

Test skal ha fokus på arbeidsprosesser og ende-til-ende test. Det skal vektlegges å verifisere at definerte arbeidsrutiner/arbeidsprosesser lar seg gjennomføre. Definerte testprosedyrer/tester skal være tilpasset definerte arbeidsprosesser og spesifiserte rutiner

Korrekthet

Funksjonalitet gir riktige eller forventet resultater iht. spesifiserte brukernes krav.

For eks. korrekthet i beregninger, riktig overføring av data mellom systemer, riktig oppdatering av data eller riktig konvertering av gamle data til nye data.

Denne kvalitetsegenskapen understøtter på mange måter kvalitetsegenskap for

Velegnethet/Suitability.

¹ ET: Enhetstest, IT: Integrasjonstest, ST: Systemtest, SIT: Systemintegrasjonstest, FAT: Factory Acceptance Test, SAT: Site Acceptance Test, PT: Produksjonstest

Samspill mellom programmoduler

Integrasjon mellom moduler i systemet fungerer som det skal.

Samspill med eksterne systemer

Integrasjon med eksterne systemer fungerer som det skal.

Samspill med annen programvare

Evne til å fungere med annen programvare, for eks. Microsoft Word eller Excel.

Sikkerhet

Systemets evne til å forhindre tilsiktet eller utilsiktet uautorisert tilgang til funksjonalitet, programmer, data eller utstyr. Dette gjelder både interne og eksterne.

Pålitelighet

Funksjonell feiltoleranse og negativ test

For eksempel, godtar feltet tomt innhold når inputverdien er obligatorisk, godtar et numerisk felt en tekst, godtar feltet et ugyldig tall, ved feil inputverdier hvordan systemet reagerer, etc.

Brukervennlighet og brukbarhet

Disse kvalitetsegenskapene burde ha en grundig test i analysefasen eller i begynnelsen av designfase, altså da man designer skjermbilder/GUI. I hvor stor grad må det testes i systemintegrasjonstesten, systemtesten og akseptansetesten kommer det an på endringsomfang.

Forståelig

Basert på skjermbilledesign, prosessflyt og terminologi/ordbruk, hvor mye innsats trengs for å forstå hva som må gjøres.

Enkel å lære

Hvor mye innsats/tid som må til for å lære å bruke systemet.

Effektiv i bruk

Hvor mye innsats/tid som må til for å fullføre for eks. en oppgave i en arbeidsprosess eller hvor mange tastetrykk/museklikk trenger brukeren for å utføre en slik oppgave.

Attraktivitet

For eksempel, er skrifttyper, farge, utseendemessig/layout, skriftstørrelser, etc. passende og konsistens fra side til side.

Samsvar

Samsvar med standarder og forskrifter

Programvaren og systemets funksjoner er i samsvar med standarder og regler fastsatt ved lover/forskrifter blant annet som sykehusets regelverk, sykehuspartners regelverk, statens regelverk eller internasjonal regelverk. Det forutsettes at disse er analysert og ivarettatt av kravspesifikasjoner.

8.2 Tekniske kvalitetsegenskaper

Følgende egenskaper kan i stor grad testes i ulike testnivåer som en del av teknisk test og utføres av personer med teknisk kompetanse.

Tabellen under lister opp tekniske egenskaper som skal testes eller verifiseres på de forskjellige testnivåene.

| Egenskaper ² | ET | IT | ST | SIT | FAT/ SAT | OAT | PT |
|--|----|----|----|-----|-------------|-----|----|
| Sikkerhet | | | | | | | |
| <i>Sikkerhet (mot uønsket adgang)</i> | X | X | X | X | X | | X |
| Pålitelighet | | | | | | | |
| <i>Robusthet: Feiltoleranse – Redundant</i> | X | X | X | X | X | X | X |
| <i>Modenhet for drift</i> | | | | | X | X | X |
| <i>Recovery</i> | | | X | | X | X | X |
| <i>Backup</i> | | | X | | X | X | X |
| <i>Samtidighet</i> | | | X | | X | | X |
| <i>Stresstoleranse</i> | | | X | | X | | X |
| <i>Tilgjengelighet</i> | | | X | | X | | X |
| Ytelse | | | | | | | |
| <i>Online responstider ved normal bruk</i> | | | X | | X | | X |
| <i>Maskinens ressursbruk</i> | | | X | | X | | X |
| <i>Tid for satsvise kjøring (batcher)</i> | | | X | | X | | X |
| Vedlikeholdbarhet | | | | | | | |
| <i>Analyserbarhet</i> | X | X | | | | | |
| <i>Forutsigbarhet</i> | X | X | | | | | |
| <i>Endringsbarhet</i> | X | X | | | | | |
| <i>Testbarhet</i> | X | X | | | | | |
| Installasjon/Driftbarhet/Flyttbarhet | | | | | | | |
| <i>Installasjon</i> | | | X | | | X | X |
| <i>Driftbarhet</i> | | | X | | | X | X |
| <i>Flyttbarhet</i> | | | X | | | X | X |
| <i>Overvåking - Alerts</i> | | | X | | | X | X |
| Teknisk infrastruktur | | | | | | | |
| <i>Software og hardware komponent</i> | X | X | X | X | X | | X |
| Samsvar | | | | | | | |
| <i>Samsvar med standarder, forskrift og teknisk arkitektur</i> | | | X | | X | | |

Tabell Oversikt over tekniske kvalitetsegenskaper

Sikkerhet

For eksempel kan en utenforstående ta en kopi av en fil, en tabell eller hele databasen, etc.

Pålitelighet

Robusthet: Feiltoleranse – Redundant

Evnen til å opprettholde et spesifisert ytelsesnivå, konsistens og berørte data ved tilfeller av defekter eller feil bruk av system eller systemstopp/avbrudd.

Årsaken kan være strømbrytning, maskinvarefeil, driftsteknikerfeil eller feil på lagringsenheter.

² ET: Enhetstest, IT: Integrasjonstest, ST: Systemtest, SIT: Systemintegrasjonstest, FAT: Factory Acceptance Test, SAT: Site Acceptance Test, OAT: Operational Acceptance Test, PT: Produksjonstest

Hvordan systemet reagerer ved feil inputparametere, feil verdier i database eller feil i operativsystem.
Hva skjer dersom enkeltkomponenter i teknisk infrastruktur slutter å fungere? Vil systemet fortsatt fungere?

Modenhet for drift

Hyppigheten av at systemet feiler på grunn av defekter i programvaren.

Recovery

Kan programvaren gjenoppta arbeid og gjenopprette data?

For eksempel

1. Mens systemet kjører, plutselig restart datamaskinen og sjekk om du har mistet det du har arbeidet med.
2. Mens systemet mottar data gjennom et nettverk, koble i fra ledningen. Etter litt tid, plugg kablet inn igjen, og analyser programmets evne til å fortsette å motta data fra det punktet hvor nettverkstilkoblingen forsvant.

Backup

Muligheten til sikkerhetskopiering av database, filer, koder, etc.

Stresstoleranse

Evnen til å håndtere unormale mengder eller frekvenser av data. Ved hvilket punkt når systemet grensen for å miste data og produsere feil, etc? Ved stress, vil systemet bare fungere langsommere eller vil det fungere feilaktig?

Denne egenskapen bør testes sammen med egenskap *Online responstider ved normal bruk og Samtidighet*.

Samtidighet

Evnen til å håndtere mange samtidige brukere. Denne egenskapen bør testes sammen med egenskap for ytelse.

Tilgjengelighet

Systemets evne til å opprettholde tilgjengelighet og kvalitet på et gitt nivå og i et gitt tidsrom.

Ytelse

Online responstider ved normal bruk

Brukeropplevd responstid i forskjellige tidsintervaller om dagen, for eks. ved rushtid der mange samtidige brukere er pålogget eller ved den tid der pågår tunge transaksjoner, osv. Denne egenskapen bør testes sammen med egenskapene *samtidighet* og *stresstoleranse*.

Maskinens ressursbruk

For eks. hvordan systemet kan bruke for mye CPU-tid, ta for stor del av minne, gi altfor mye nettverkstrafikk eller bruke altfor mye diskplass.

Tid for satsvise kjøring (batcher)

Kjøringstid til batchfunksjoner/batchprogrammer.

Vedlikeholdbarhet

Denne egenskapen testes i større grad under design/oppbygging av koden. Den testes og verifiseres dermed stort sett av utviklere under utviklingsarbeid, enhetstest og integrasjonstest.

Analyserbarhet

Innsatsen som trengs for å finne defekter eller årsaken til feil i koden. For eks. kode som er standardisert, modularisert, strukturert og godt dokumentert er kilder for analyserbarhet.

Forutsigbarhet

Risiko for kodeendringer har uventede effekter.

Endringsbarhet

Innsatsen som trengs for å endre kode og omgivelsene.

Testbarhet

Innsatsen som trengs for å teste systemet etter endringer.

Flyttbarhet/Installasjon/driftsbarhet

Installasjon

Innsatsen som trengs for å installere systemet i spesifiserte omgivelser. Det skal vektlegges å verifisere at definerte installasjonsrutiner lar seg gjennomføre og systemet er klart for bruk. Det omfattes også installasjonstest av software og hardware komponenter i teknisk infrastruktur.

Driftsbarhet

Innsatsen som trengs for å drifte systemet i spesifiserte omgivelser. Det skal vektlegges å verifisere at definerte driftsrutiner lar seg gjennomføre.

Flyttbarhet

Mulighetene for å tilpasse systemet til forskjellige spesifiserte omgivelser uten andre midler enn de som er innbygd i systemet. Hvor for eksempel den ene omgivelsen har Linux versjon D mens den annen har Linux C.

Man kan for eksempel foreta en verifikasjon av denne egenskapen i akseptansetestmiljøet etter flytting av systemet fra systemtestmiljøet, eller i produksjonsmiljøet etter flytting av systemet fra akseptansetestmiljøet.

Overensstemmelse med standarder og konvensjoner som vedrører flyttbarhet bør også verifiseres.

Overvåking - Alerts

Mulighetene for å overvåke systemet og varsle når feil oppstår i systemet.

Teknisk infrastruktur

Software og hardware komponent

Software og hardware komponenter testes i forhold til overnevnte tekniske kvalitetsegenskaper.

Samsvar

Samsvar med standard, forskrifter og teknisk infrastruktur

Programvaren og teknisk arkitekturen er i samsvar med standarder, regler og teknisk infrastruktur fastsatt ved lover/forskrifter blant annet som sykehusets regelverk, sykehuspartners regelverk, statens regelverk eller internasjonal regelverk. Det forutsettes at disse er analysert og ivaretatt av teknisk krav.

9 Testtyper og testteknikker

Det må velges hensiktsmessige testtyper og teknikker for de ulike testnivåene.

9.1 Testtyper

De to vanligste testtypene er “Black box” testing og “White box” testing. Black benyttes vanligvis i systemtest, systemintegrasjonstest, akseptansetest og produksjonstest, mens Whitbox benyttes stort sett i enhetstest og integrasjonstest.

Black box testing er basert på kravspesifikasjon, med fokus på å verifisere at testobjektets resultat er korrekt ift. brukernes spesifiserte behov. Dvs. Man tester på applikasjon basert på for eks det som viser fram på skjermbilder, uten å ha kjennskap til hvordan programkoden er teknisk bygget opp.

White box testing er basert interne programkodens strukturer, med fokus på å verifisere at enkelte strukturer fungerer som det skal. Dette blir utført av systemutviklerne.

9.2 Testteknikker

Hensikten med testdesignteknikker er for å gi riktig fokus i testgjennomføring og design av testcase, for å oppnå den forventet testdekningsgrad av de ulike testobjekter spesifisert i risikoanalyse.

Spesifikasjonsbaserte teknikker (svart boks)

- Brukstilfeller/brukerscenarier
Tester blir spesifisert/definert på bakgrunn av brukernes behov som er for eks. beskrevet i kravspesifikasjoner eller løsningsbeskrivelser
- Ekvivalensklasseinndeling
- Grenseverdianalyse
- Beslutningstabell
- Tilstandsbasert
- Destruktiv
- Negativ testing
Å bevisst utføre feil for å undersøke hvordan løsningen håndterer ulike feilsituasjoner.

Erfaringsbaserte teknikker (svart boks)

- Eksplorativ testing (utforskende testing, fritest, adhocstest)
Denne tilnæringsmåten er basert på en testoppgave som inneholder testmål og som utføres i fast bestemte tidsintervaller. Den er mest nyttig når man har få eller mangelfulle spesifikasjoner og meget stort tidspress, eller for å supplere en formell testing. Den kan fungere som en bra verifisering for å sikre at de mest alvorlige feil blir funnet.
- Feilgjetting
En vanlig brukt erfaringsbasert teknikk er feilgjetting. Generelt forutser testere feil basert på erfaring. En strukturert tilnæringsmåte til feilgjettingsteknikken er å spesifisere en liste av mulige feil og lage testtilfeller som angriper disse feilene. Denne systematiske tilnæringsmåten kalles fault attack.

Strukturertbaserte teknikker (hvit boks)

- Dekning
- Statement
- Sti
- Funksjon
- Tilstand

10 Inngangs- og utgangskriterier

For å sikre at alle kvalitetssikringsaktiviteter er gjennomført på tidligere testnivå er det viktig å verifisere at inngangskriteriene til et testnivå og/eller en testaktivitet er oppfylt. I tabellen under vil et sett med generelle kriterier bli listet opp sammen med utgangskriterier. Disse må tilpasses hvert testnivå/testaktivitet i den enkelte testplan.

| Type kriteria | Beskrivelse |
|-------------------|--|
| Inngangskriterier | Dokumentasjon fra foregående testnivå er levert og godkjent. Installasjonstest i testmiljøet er gjennomført og godkjent. Testplan er godkjent. Testprosedyrer er kvalitetssikret og klare for gjennomføring. Testdata er klargjort. Opplæring gjennomført. Testere er bestilt. |
| Utgangskriterier | Testdekning er godkjent Avvik fra plan er dokumentert og risikovurdert Testrapport er ferdigstilt |

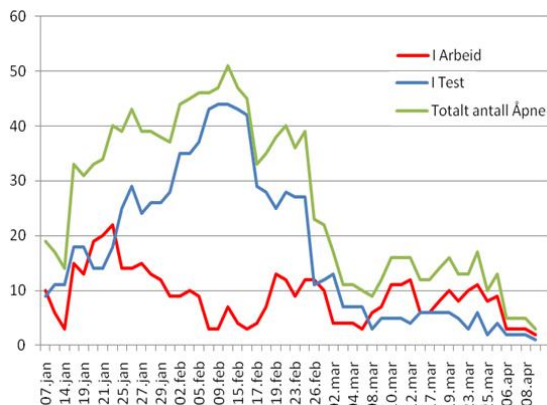
Tabell Inngangs- og utgangskriterier

11 Godkjenningskriterier

Kriterier for godkjenning av test skal beskrives for den enkelte test i testplan og må forankres hos prosjektledelsen. Punktene listet opp nedenfor er generelle krav som er gjeldene for test hvis ikke annet er beskrevet i testplan. Dette gjelder for alle testnivåer.

Vurdering av godkjenningskriterier for at testen kan godkjennes:

1. I utgangspunktet skal alle planlagte tester være gjennomført.
2. Krav til dekningsgrad skal være innfridd.
3. Alle observerte feil er dokumentert.
4. Gjenstående antall alvorlige avvik eller små avvik skal være akseptert eller i henhold til kontrakt som er gjengitt i testplan.
5. Retteplan er utarbeidet for utestående feil.
6. Vise til nedadgående feiltrend ("antall feil/testtime", e.l. mål) i testen.
7. Det skal være utarbeidet statusrapport med anbefaling om godkjenning eller underkjenning av testnivået.



Synkende feilrate

Antall feil skal være synkende mot slutten av testen. Et system som ikke er stabilt har en feilrate som ennå stiger, eller synker og øker igjen - samtidig som testintensiteten er tilnærmet den samme. Dette forteller at systemet ikke har god nok produktkvalitet til å produseres. Med en urolig feilrate er det større risiko for kritiske feil som stopper hele eller de mest forretningskritiske delene av systemet.

12 Avbrudds- og gjenopptagelseskriterier

Stopp av test er en alvorlig hendelse og skal derfor nøye vurderes om det er nødvendig. Testen skal stoppes når:

- Det avdekkes et vesentlig antall feil som burde vært funnet i tidligere testfase. Tidligere testfase analyseres og kompletterende tester gjennomføres
- Det oppdages manglende kontroll med testmiljø eller testdata
- Det oppdages manglende kontroll med konfigurasjonsstyring
- Det avdekkes feil som gjør det umulig å fortsette testen
- Det avdekkes så mange feil at kravet åpenbart ikke vil bli akseptert
- Etterslep i feilretting medfører at det ikke er mulig å teste mot siste versjon
- Det avdekkes alvorlige mangler eller feil i testbasis og/eller testdokumentasjon

Testleder har myndighet til å stoppe testen, men dette skal umiddelbart rapporteres til delprosjektleder test eller prosjektleder.

Testen kan gjenopptas når årsaken til avbruddet er avdekket og utbedret. Det bør vurderes om testen skal utvides med tilsvarende antall dager som testen ble stoppet.

Det vil være både test- og feilavhengig om testen skal gjenopptas der den ble avbrutt eller om hele testfasen skal gjentas. Det vil være opp til en konkret vurdering i det enkelte tilfelle på hvilke sted i prosessen testingen skal gjenopptas, og dermed hvilke deler av testen som skal gjentas etter at feilen eller feilene, er rettet. Denne vurderingen tas av testleder i samråd med delprosjektleder test eller prosjektleder.

13 Testdokumentasjon og rutiner

Hensikten med testdokumentasjonen er:

- Den skal støtte en effektiv testprosess
- Den skal videreformidle den informasjon man får ut av testing

Nedenfor er det beskrevet hvilke dokument som skal utarbeides for hvert testnivå:

| Dokument | Beskrivelse |
|--|---|
| Teststrategi | Det kan utarbeide en teststrategi som tar utgangspunkt fra Sykehuspartners teststrategi. Eller lage en overordnet/master testplan og bruke Sykehuspartner teststrategi som basis. Det anbefales bruk av Sykehuspartners mal til teststrategi. |
| Testplan | Ved større prosjekt må overordnet/master testplan utarbeides. Det skal utarbeides en detaljert testplan for hvert testnivå. For mindre prosjekt kan en testplan inneholde flere testnivåer. Det anbefales bruk av Sykehuspartners mal til testplan. |
| Testbeskrivelser i HP ALM | Detaljert beskrivelse av hver enkel test i HP ALM Testbetingelser Forventet resultat Faktisk resultat Testprosedyrer |
| Krav/løsningsbeskrivelse i HP ALM | Testprosjektet må ta stilling om krav/løsningsbeskrivelse bør legges inn i HP ALM. Fordelen er at testdekning kan sjekkes automatisk når krav er lenket med tester. Ulempe er at krav også ligger på et annet sted, ved oppdatering må det oppdateres på flere steder. Alternativet er å kunne legge inn navn (overskrift) til krav i HP ALM, men ikke innholdet. |
| Avvik/feil i HP ALM | Registrere og oppdatere i HP ALM |
| Testdata | Dokument som beskriver de testdata som skal benyttes i aktuell testnivå |
| Testrapport/sluttrapport/ framdriftsrapport | Testrapport Det anbefales bruk av Sykehuspartners mal til testrapport |
| Evaluering | Evaluering |

All dokumentasjon som ikke ligger i HP ALM må lagres elektronisk. Dersom det lagres testdokumentasjon i forskjellige verktøy skal det lages en henvisning til hvor den finnes. Navn og versjonshåndtering er viktig for enkelte å kunne finne igjen dokumenter.

Det anbefales bruk av Sykehuspartners testrelaterte rutiner som ROS-analyse, HP ALM brukerveiledning, rollebeskrivelser, etc.

14 Krav og forutsetninger til testmiljø og fysiske omgivelser

14.1 Testmiljø

Testmiljøet skal beskrives i testplanen for leveransen/testnivået.

Det er normalt at eget testmiljø opprettes for test av infrastruktur og ikke funksjonell test, og eget testmiljø for funksjonell test. Test av datakonvertering utføres også ofte på eget miljø. Jo høyere testnivå det er på testen, jo mer lik produksjonsmiljøet må testmiljøet være. Avvik mellom testmiljøet og produksjonsmiljøet må dokumenteres i testplanen. Følgende må vurderes ifht. avvik:

- Maskinvare – servere, nettverk, linjer etc
- All programvare – 3dje part
- Testdata
- Klienter og skrivere
- Sikkerhetsmodell/regime

Leveranser til testmiljøet skal skje kontrollert og skal godkjennes av testleder. Dersom flere prosjekter jobber mot samme testmiljø, skal leveransene koordineres. Alle endringer i testmiljøet skal dokumenteres og begrunnes.

14.2 Fysiske omgivelser

Hvilke lokaler og antall arbeidsplasser som skal benyttes i test, beskrives i detaljerte testplaner for det enkelte prosjekt. Det bør prioriteres å samle testerne i et eget rom/egen gruppe når testene skal gjennomføres.

15 Testdata

Krav til testdata i hvert enkelt prosjekt skal dokumenteres i testplanen.

Testdata skal inneholde et representativt grunnlag i forhold til produksjon og de testene som skal gjennomføres. De velges på en slik måte at de gir en god testdekning for de funksjonelle områdene som skal testes. Hvilke testdata som benyttes skal dokumenteres enten direkte i testscenariet eller i et eget skjema.

Det er viktig å være oppmerksomme på eventuelle krav fra Datatilsynet. I en testsituasjon vil testerne normalt ha tilgang til testdata utover det de har tilgang i produksjon. Hvis dette er tilfelle må testerne underskrive en taushetserklæring.

Der det er praktisk mulig skal testdata anonymiseres.

15.1 Sikkerhet ifb. med midlertidige datalagringsenheter

Lagringsenheter som benyttes i ulike testmiljø inneholder produksjonsdata må oppbevares i samsvar med sikkerhetskravene.

16 Roller, organisering og ansvar

Roller og ansvar beskrives detaljert i den enkelte testplan og Sykehuspartners rollebeskrivelser for testgjennomføring. En person kan ha flere roller, avhengig av prosjektets størrelse. Følgende er oversikt over roller som er anbefalt å ha i testprosjektet.

- Delprosjektleder
- Testleder (funksjonell og/eller teknisk)
- Testere (funksjonell og /eller teknisk)
- Defect manager
- Testkoordinator
- Testmiljø - Driftsansvarlig
- Testverktøysansvarlig

16.1 Bemannings og opplæringsbehov

Bemanningsbehov defineres i testplanen for hvert prosjekt. Det er behov for funksjonell opplæring av alle testere før testbetingelser, testscenarier etableres og test gjennomføres.

I funksjonell akseptansetesten, systemintegrasjonstesten og systemtesten vil sluttbrukere bli benyttet til test av systemet. Som sluttbrukere menes personell fra

- Helseforetakene
- Forvaltningsorganisasjonen
- Support/Brukerstøtte (1.linje og 2. linje support/brukerstøtte)

Det enkelte prosjekt vurderer behov for – og gjennomfører opplæring i:

- Generelt testarbeid (fokus for testarbeid, holdninger, etc)
- Testverktøyet/testprosedyrer
- Systemet som skal testes
- Bruk av feilregistreringsverktøy

17 Avvikshåndtering

Alle observasjoner av forhold som antas å representere en feil skal registreres i testverktøyet HP ALM.

Testansvarlig har ansvaret for at innkomne observasjoner vurderes og klassifiseres, samt behov for eventuell retest og regresjonstesting.

En observasjon er ikke en feil dersom den oppsto på grunn av:

- feil i testbeskrivelser/testcase
- misforståelse hos tester

Observasjonene kan henlegges hvis problemet som oppsto:

- ikke er feil i forhold til endringsmelding/praksis
- skyldes feil som henlegges etter avtale (ut fra kost/nytte vurdering)
- ikke kan reproduseres

Finner tester feil i akseptansetest skal følgende vurderes:

1. Lar feilen seg rette raskt slik at ny testing kan gjennomføres?
2. Hva er konsekvensene ved å trekke ut enheten og evt. andre enheter som er avhengig av denne?
3. Hva er konsekvensene av å produsjonssette systemet med feil? Kan manuelle rutiner etableres som en midlertidig løsning?
4. Kan feilen produsjonssettes uten at det medfører problemer for andre deler av systemet?

17.1 Alvorlighetsgrad av avvik

Følgende alvorlighetsgrad anbefales.

| | Alvorlighetsgrad | Beskrivelse |
|---|-----------------------------|---|
| A | Kritisk feil | Feil som medfører at standardsystem og tilpasning stopper, at data går tapt eller at andre vesentlige funksjoner for kunden ikke er levert eller ikke virker som avtalt. Feil ved dokumentasjon som gjør at kunden ikke kan bruke standardsystem og tilpasning eller vesentlig deler av den som avtalt. |
| B | Alvorlig feil | Feil som medfører at funksjoner som er viktige for kunden ikke virker som beskrevet i avtalen og som det er tids og ressurskrevende å omgå. Dokumentasjon som er viktig for kunden er mangelfull, upresis eller lett kan misforstås |
| C | Mindre alvorlig feil | Feil som medfører at en enkeltfunksjon ikke virker som avtalt, men som kunden relativt lett kan omgå. Dokumentasjonen er mangelfull, upresis eller kan lett misforstås. |
| D | Ikke alvorlige feil/Småfeil | Ubetydelige feil f. eks. i brukergrensesnittet. Trykkfeil, formateringsfeil samt andre småfeil i dokumentasjon som ikke påvirker forståelsen av dokumentasjonen. |

17.2 Status avvik i HP ALM

Følgende statusene anbefales.

| Status | Beskrivelse |
|----------------|---|
| New | Ny feil legges inn i HP ALM |
| Open | Feilen er sett på/behandlet, men fremdeles åpen |
| Rejected | Feilen er avvist, det er ingen feil |
| Closed | Feilen er lukket/avsluttet |
| Pending | Venter på at feilen blir rettet |
| Fixed | Feilen er rettet |
| Retest | Ny versjon er klar for å testes på nytt |
| Reopen | Feilen ble ikke rettet og defecten må åpnes på nytt |
| Exists in prod | Feilen er kjent og eksisterer i produksjon i dag |

17.3 Endringshåndtering

Endringer registreres underveis i testarbeidet. Det må avklares om endringer tas med eller ikke tas med som en del av leveransen.

Følgende tilfeller er endringsønsker:

- Egenskaper som er beskrevet i designdokumentasjonen eller kravspesifikasjonen og som fungerer i henhold til denne, men hvor kunden ønsker en annen løsning.
- Egenskaper i systemet som fungerer, men hvor detaljer om virkemåte ikke er beskrevet i designdokumentasjonen eller kravspesifikasjonen og hvor kunden ønsker en annen løsning.
- Nye egenskaper i systemet som kunden ønsker og som ikke er spesifisert i designdokumentasjonen eller kravspesifikasjonen.

18 Testverktøy

Sykehuspartner har valgt å ta i bruk testverktøy *HP Application Lifecycle Management (HP ALM)* for administrasjon av testing og tester.

18.1 Verktøy for administrasjon av testing og tester

HP ALM er et komplett testverktøy som understøtter hele testprosessen. Det vil si testplanlegging, gjennomføring, oppfølging av avvik, oppfølging av progresjon og kvalitet samt rapportering. Verktøyet gir testansvarlig oversikt og rapporteringsmuligheter, sikrer sporbarhet og mulighet til enhetlig dokumentasjon av test- og feilrettingsaktivitetene. ALM benyttes for alle testnivåer. Det skal også benyttes i sammenheng med kravhåndtering, vurdering av testdekning, utvikling av testprosedyrer/testcase, testplaner og feilhåndtering.

Det skal opprettes et eget prosjekt i *HP ALM* for testprosjektet.

Som minimum skal alle planlagte tester kjøres minst en gang, eller det skrives en avvikrapport dersom tester ikke kan gjennomføres. Testene basert på krav med høy risiko kjøres i regresjoner angitt i detaljert testplan. Tester forbundet med høyere risiko bør regresjonstestes uansett endring på systemet og typisk siste gang i akseptansetestperioden

Det er ferdig utarbeidet brukerveiledninger og e-læringskurs til *HP ALM* for både testere og testledere.

18.2 Verktøy for automatisert funksjonell test

Sykehuspartner jobber med vurdering av ulike potensielle verktøy, deriblant *HP QTP*, etc.

18.3 Verktøy for automatisert teknisk test

Sykehuspartner jobber med vurdering av ulike potensielle verktøy for ytelsestest, deriblant *HP LoadRunner*, *AppLoader*, etc.

For test av infrastruktur har Sykehuspartner tatt i bruk blant annet *IPERF/JPERF*, *Chariot*, *PinpPlotter*, *Multicast Hammer*, *Spirent*, *SwingBench*, etc.

Se teststrategi for ikke funksjonell og teknisk infrastruktur test.

18.4 Testautomatisering

Det anbefales å vurdere muligheten for automatisert testing både for tekniske tester og funksjonelle tester i oppgradering, innføring og utviklingsprosjekter

Automatisert testing er et viktig supplement, men kan ikke alene erstatte den manuelle testingen. Per dags dato eksisterer det ingen automatiserte tester som dekker 100 % av arbeidet til testerrollen som en person har. Disse automatiserte testene benyttes til å ta bort den ofte kjedelige og repetitive testingen slik at testressurser kan fokusere på den mer utfordrende delen av testingen.

Automatisert testing egner seg p.t. i stor grad for tekniske tester hvor for eks. testverktøy for ytelsestest har mulighet til å simulere tusenvis av fiktive brukere som kan samtidige utføre ulike funksjoner/oppgaver på systemet i løpet av en beregnet tid. En slik mulighet vil neppe få til med manuelle tester på en effektiv måte.

I utgangspunktet er automatisert testing egnet for gjentakende testing av scenarier som i liten grad endrer seg. Fordeler ved automatisk testing vil blant annet være:

- Hurtigere gjennomføring av tester med færre personer
- Gjenbruk av testscript

- Rask verifikasjon mot forrige kjøring
- Det kan testes med flere datavariasjoner, større datamengder og større antall brukere enn det er mulighet for ved manuell testing
- Eliminerer menneskelig feil i gjennomføring

Typiske kriterier for scenarier som egner seg for automatisk testing:

- Defineres en gang og skal kjøres mange ganger
- Grensesnitt det testes mot er stabilt mellom kjøring
- Scenariet er stabilt mellom kjøring
- Kan sammenlikne resultater fra en kjøring med tidligere kjøring

19 Generelle prinsipper om testing

Testing skal følge prinsippene under:

1. Å planlegge, tilrettelegge og gjennomføre systematisk og metodisk testarbeid er en av de beste teststrategiene.
2. Testplanleggingen startes så tidlig som mulig i prosjektløpet.
3. Enhver endring av it-løsningene skal være gjenstand for test.
4. All test skal være risikobasert. Risikoanalysen avgjør testomfanget, testdekningsgrad, hvilke testnivåer, hvilke testteknikker og kvalitetsegenskaper skal benyttes i hvilke testobjekter. Den avgjør også omfanget til regresjonstesting.
5. Retest skal alltid benyttes på en hver utført feilretting.
6. Alle testaktiviteter skal være dokumentert slik at sporbarhet i hele testprosessen sikres.
7. Teststrategien skal brukes som referansedokument ved inngåelse av kontraktuelle betingelser med leverandør slik at krav i denne strategien kan ivaretas. Dokumentet skal brukes som retningslinjer for utarbeidelse av testplan og testgjennomføring.
8. Alle kvalitetsegenskaper skal vurderes testet både positivt og negativt.
9. Test skal starte så tidlig som mulig.
Kostnaden med å rette feil øker eksponentielt fra testnivå til testnivå jo nærmere produksjonssetting man kommer. I produksjon kan feilkostnader være store. Test skal bidra til å finne feil tidligst mulig.
10. Testing må tilpasses feilområdet.
Feil er ikke likt fordelt. Økt testdekningsgrad og testinnsats på områder der det skjer flest feil.
11. Feil blir etter hvert ”immune” av samme test. Gjentakende like tester bidrar lite til å finne nye feil, men er nødvendig for å sikre at tidligere feil ikke er gjeninnført og at de faktisk er rettet.

Andre testteknikker bør tas i bruk, eksempelvis erfaringsbaserte teknikker i stedet for spesifikasjonsbaserte teknikker.
12. QA gjennomgang er en vesentlig aktivitet som kan bidra til å finne feil tidlig i kjerne dokumenter i prosjektløpet.
13. Test med testdata av høyest mulig kvalitet.
Alle tester må gjøres med testdata av høy kvalitet, eller rettere sagt, med testdata som er ”reelle”, altså slik de vil fremstå i produksjon (uten at det nødvendigvis er produksjonsdata).

Vedlegg 1 Begreper og forkortelser

Begreper

| Begrep | Hensikt/beskrivelse |
|---------------------------|--|
| Akseptansekriterie | Beskrivelse av krav til systemet som må være oppfylt for at kunden/ systemeier skal akseptere systemet. Ref. godkjenningkriterier. |
| Akseptansetest (AT) | (Bruker- og akseptansetest). En formell test for å avgjøre om akseptansekriteriene er oppfylt og gi systemeier anledning til å akseptere eller avvise systemet. |
| Avvik | Fravikelse fra forventet resultat ved bruk av programvare. |
| Black Box test | Testmetode med fokus på test av funksjonalitet, krever kunnskap om kravspesifikasjonen og avtaler. |
| Driftstest | En driftstest er en test som gjennomføres for å verifisere at driftsrutinene er gode nok. Testene utformes med henblikk på å prøve ut både maskinelle og manuelle driftsrutiner, samt driftsdokumentasjon. |
| Dynamisk test | Testen som krever kjøring av programvaren for å finne feil. |
| Egenskap | Standarden ISO/IEC 9126 definerer 6 egenskaper som til sammen utgjør kvalitet for programvare. Disse egenskapene benyttes i dette dokumentet for å beskrive egenskaper for de testobjektene som skal testes |
| Endring | Endring i forhold til vedtatte/godkjente krav, til funksjonalitet, til arbeidsflyt, til infrastruktur, til sikkerhetspolicy, til dokumentasjon osv. Nesten enhver endring skaper behov for test av ulikt omfang. |
| Enhetstest (ET) | Enhetstesting (eng: unit testing) er en testmetode hvor enkeltkomponenter testes hver for seg.. De enkelte testobjekt testes uavhengig av andre testobjekt eller av grupper av relaterte enheter. |
| Feil | Utslag av en handling som gjør at testobjektet produserer et uriktig resultat. |
| Funksjonell egenskap | Funksjonell egenskap karakteriserer hva programvaren omfatter for å dekke behov. |
| Godkjenningkriterier | Beslutningsregler for å avgjøre om et testobjekt eller egenskap er godkjent i testen. |
| Hendelseslogg | I denne loggen skal de timer en benytter til test hver dag registreres. Dette er Testleders "Dagbok". Det er meget viktig at alle hendelses som medfører til stopp i planlagt utført arbeid blir registrert, dette kan være slikt som at testmiljø er nede, testressurser ikke er tilgjengelig, programvare ikke levert etc. |
| Ikke-funksjonell egenskap | Ikke-funksjonell egenskap karakteriserer når og hvordan behovene dekkes. |
| Installasjonstest | En installasjonstest er en test for å verifisere om installasjonen lar seg gjøre i henhold til installasjonsdokumentasjon, at SW-leveransen er i henhold til krav, samt at den ikke forstyrrer andre applikasjoner på plattformen. Test for å verifisere at systemet kan installeres som forutsatt |
| Integrasjonstest (IT) | Test av samspillet mellom systemets deler (interne grensesnitt) og samspillet med andre systemer som det skal virke sammen med (eksterne grensesnitt). Testing med hensikt å avdekke feil i måten integrerte systemer samspiller, med hensyn til å kalle hverandre korrekt, utføre komplette behandlings-sekvenser og oversende og lagre data korrekt. |
| Konfigurasjonsstyring | Identifisere, kontrollere og organisere endringer i systemer. |
| Kvalitetsegenskap | Et produkts totale egenskaper og karakteristika som vedrører dets evne til å tilfredsstillte fastsatte og underforståtte behov. (ISO 8402:1986) |
| Pilot | Pilot er en produksjonskjøring i begrenset omfang med et forsterket og dedikert støtteapparat. I Pilot inngår utrullings- og produksjonssettingsmekanismene samt alle maskinelle - og manuelle rutiner i daglig produksjon og skal skape trygghet for at den totale løsningen er moden for utrulling. |
| QA gjennomgang | Granskning, gjennomgang av dokumentasjoner for å finne feil. Se også Statisk test. |
| Regresjonstest | Test av hele eller utvalgte deler av systemet for å verifisere at feilrettinger eller endringer ikke har introdusert nye feil eller uønskede effekter, og at systemet eller funksjonen fortsatt tilfredsstillter spesifiserte krav. |
| Retest | Testing som har til hensikt å kontrollere at en feil er fjernet. |
| Risikobasert test | Er basert på en gjennomført risikoanalyse for å sikre at områder med høy risiko blir fokusert i testprosjektet. |
| Sikkerhetstest | Test som har til hensikt å avdekke svakheter i sikkerheten til systemet. Sikkerhetstesting gjøres for å sjekke den tekniske sikkerhetstilstanden til et nettverk, applikasjon eller maskin. Formålet er å sjekke hvor sårbart denne er for eksterne eller interne angrep. Med angrep menes forsøk på å endre, slette eller kopiere beskyttelsesverdig informasjon. |
| Stabilitetstest | Stabilitetstest er en test for å måle stabiliteten på applikasjonen under normal bruk over en gitt periode, perioden må være definert med en startdato og sluttdato eller starttid og slutt-tid. Dette for å kunne sammenligne stabilitet i gitte perioder. |
| Statisk test | Granskning, gjennomgang av dokumentasjoner for å finne feil. Se også QA gjennomgang. |

| Begrep | Hensikt/beskrivelse |
|--------------------------|---|
| Stresstest | Test av systemets oppførsel med stor belastning av trafikk. Normalt vil stresstest fokusere på yttergrensene for hva systemet tåler, eller å finne disse. |
| Systemtest (ST) | En formell test som avklarer om et system virker som beskrevet i krav spesifikasjonen. Kan innbefatte spesielle ikke-funksjonelle tester for å få testet i følge kravspesifikasjonen/ forretningsprosesser for eksempel volumtest, stresstest, ytelsestest, sikkerhetstest, installasjonstest, 'online help'/brukermanual test etc |
| Teknisk test | Test av ikke-funksjonelle egenskaper som f.eks. test av stabilitet, volum, kapasitet, belastning, robusthet, svartider,,stress og ytelse. |
| Test av brukervennlighet | Avdekke forskjeller mellom brukers forventninger til systemet og faktisk funksjonalitet |
| Test rapport | Sluttrapport som oppsummerer testprosjekt, testaktiviteter og resultater. Inneholder også en evaluering av de tilhørende testobjekter. |
| Testbasis | Testbasis er all dokumentasjon som benyttes i spesifiseringsarbeidet. |
| Testbetingelser | Utdyper <i>hva</i> som skal testes og utledes av spesifikasjonen (test condition). Eks: hvis <i><betingelser></i> , så <i><hendelse/ tilstand></i> |
| Testdata | Oversikt over hvilke inputdata som skal anvendes i et testcase. Kan være kopi av reelle data eller fiktive data opprettet kun til spesielle tester. |
| Testdekning | I hvilken grad en gitt test eller samling av tester verifiserer alle spesifiserte krav for et gitt system eller komponent. |
| Testfase | De faser i et prosjekt som skal gjennomføres iht. testmodellen. Defineres i overordnet teststrategi. |
| Testgrunnlag | Den dokumentasjonen som testen blir basert på, for eksempel krav- eller designspesifikasjon. Dersom et dokument bare kan endres gjennom formell endringskontroll, er testgrunnlaget frosset. |
| Testlogg | Dette er den manuelle loggen som erstatter eller utfyller HP-ALM for logging av alle hendelser og observasjoner underveis i testløpet. |
| Testmaterieil | Alle gjenbrukbare leveransedokumenter fra test, som f.eks. teststrategier, testplaner, testbetingelser, testcase, krav til testdata, testrapporter. Det kan være et krav at hele eller deler av testmateriellet skal gjenbrukes og være mulig å vedlikeholde. |
| Testmiljø | Testmiljø er det miljøet hvor testen skal foregå, både teknisk og fysisk. Det er vesentlig at dette miljøet er så nært opp til speilbilde av det eksisterende eller fremtidige produksjonsmiljøet. |
| Testnivå | De nivåer/avgrensede tester som et system/endringer bør testes gjennom. Nivåene som skal brukes, tilpasset til gjeldende system, spesifiseres i teststrategien. I denne modellen brukes begrepene enhetstest, systemtest, og akseptansetest om testnivåer. Innenfor hvert enkelt nivå er det et sett med anbefalte testtyper |
| Testobjekt | En programvarekomponent eller en samling av programvare som er gjenstand for testing. Testobjekt i denne testplanen defineres som dokumenter og programmer som skal gå i drift. Den minste testbare enhet. Størrelsen på enheten avhenger av hvilken testfase som gjennomføres. |
| Testplan | Dokument som beskriver hensikt, omfang, fremgangsmåte, organisasjon, ansvarsforhold, fremdriftsplan for planlagte testaktiviteter og godkjenningkriterier. |
| Testscenario/ Testcase | Definerer et sett av inndata og forventet resultat av aktuelt case. En testcase definerer <i>hvordan</i> testbetingelsene skal testes og viser hvilke trinn som skal gjennomføres for å kjøre testen. Testcase skal alltid inneholde hvilke testdata som skal benyttes og forventet resultat. Dette kan f.eks. gjøres ved at det utarbeides egne vedlegg med testdata som knyttes til testscenarioet. |
| Teststrategi | Et dokument som overordnet beskriver testarbeidet (testprosessen/testprosjektet) og de ansvarsforhold som gjelder i et prosjekt. Det skal minimum beskrives hvilke testnivåer som skal gjennomføres samt hvilke egenskaper som det skal fokuseres på, testobjekter, dekningsgrad, roller og ansvarsforhold, fremdriftsplan og begrepsapparat. |
| Testtype | Det finnes mange typer test, og valg av type baseres på hva som skal testes og hensikten med aktuell test. Eks.: Regresjonstest, brukervennlighetstest, ytelsestest, stresstest, volumtest, sikkerhetstest osv. Testtyper kan utføres på flere testnivå. |
| Testverktøy | Programmer og verktøy som benyttes til hjelp ved testing, for eksempel planlegging av tester, oppfølging av avvik, generering av testdata, simulering av belastning på maskinvare osv. |
| Volumtest | Volumtest er en test av systemets oppførsel ved prosessering av store datamengder. |
| White box test | Er en testmetode som går på testing av kjente strukturer. Det tas hensyn til programmets interne struktur, logikk og behandlingsregler. |
| Ytelsestest | Ytelsestest er en test for å måle svartider under gitte bruksmønstre i applikasjonen og systemets evne til prosessering under gitte krav til svartider. |

Forkortelser

| Forkortelse | Beskrivelse |
|-------------|-----------------------------|
| AT | Akseptansetest |
| DPL | Delprosjektleder |
| FAT | Factory Acceptance Test |
| ET | Enhetstest |
| HSØ | Helse Sør-Øst |
| HF | Helseforetak |
| IT | Integrasjonstest |
| OAT | Operational Acceptance Test |
| PL | Prosjektleder |
| PT | Produksjonstest |
| RHF | Regionalt Helse Foretak |
| SAT | Site Acceptance Test |
| SIT | Systemintegrasjonstest |
| SP | Sykehuspartner |
| ST | Systemtest |
| TL | Testleder |
| TST | Tester |