

# Overordnet Testplan

---

*MUSIT Ny IT-arkitektur, Pilot og Hovedprosjekt*

# Innhold

1	Innledning .....	4
1.1	Hensikten med dette dokumentet .....	4
1.2	Grensesnitt .....	4
1.3	Omfang av dokumentet .....	4
1.4	Definisjoner .....	4
2	Overordnede prinsipper.....	6
2.1	Gradering av avvik .....	6
2.2	Nettlesere .....	6
2.3	Mobile plattformer .....	7
3	Gjennomføring av test.....	8
3.1	Minimumstest .....	8
3.2	Feilretting i testperioden .....	8
3.3	Risikobasert testing .....	8
3.4	Sprinttest .....	8
3.4.1	Startkriterier .....	8
3.4.2	Gjennomføring.....	9
3.4.3	Utgangskriterier .....	9
3.5	Integrasjonstest.....	9
3.6	Systemtest - Funksjonalitet .....	9
3.7	Ytelsestest.....	9
3.8	Stresstest .....	9
4	Testmiljø .....	10
5	Akseptanse og Godkjenning .....	11

# Versjonshistorikk

Versjon	Dato	Endret av	Godkjent av	Endringer
0.9	10.06.2016	Line Arild Sjo		Dokument klart for godkjenning i styringsgruppen
1.0	15.06.2016	Line Arild Sjo	Styringsgruppen	Dokumentet oppdatert med kommentarer fra styringsgruppen

## Relaterte dokumenter

Ref. nr.	Dokument	Lenke
1	Styringsdokument for MUSIT Ny IT-arkitektur, Pilot	
2	Styringsdokument for MUSIT Ny IT-arkitektur, Hovedprosjekt	
3	Tjenesteoversikt GAP-analyse	

# 1 Innledning

## 1.1 Hensikten med dette dokumentet

Dette dokumentet beskriver forutsetninger for og gjennomføring av test av Pilot og Hovedprosjektet for Ny IT-arkitektur hos MUSIT. Det henvises til *Styringsdokument for MUSIT ny IT-Arkitektur, Pilot* og *Styringsdokument for MUSIT ny IT-Arkitektur, Hovedprosjekt* for overordnede prinsipper og forklaringer.

Overordnet testplan omhandler testing av alle leveransepakker, og har til hensikt å:

- synliggjøre hva som skal testes, hvordan testingen skal gjennomføres, og hvilket testmiljø testfasene skal gjennomføres i.
- beskrive ansvarsforhold relatert til forberedelser, gjennomføring og godkjenning av de ulike testfasene.
- legge forholdene til rette for at testaktiviteten blir gjennomført på en effektiv måte, slik at hovedutfordringer ivaretas på best mulig måte samt at så mange feil som mulig avdekkes så tidlig som mulig.

Et annet viktig formål med overordnet testplan er å danne grunnlaget for planleggingsarbeidet som skal gjennomføres. Overordnet testplan søker således å synliggjøre og beskrive aktiviteter som er styrende for hvordan videre planleggingsarbeid i utgangspunktet innrettes.

## 1.2 Grensesnitt.

Grensesnittene vil bli definert som del av kravspesifiseringen av de ulike modulene (delleveransene). Test av disse vil beskrives i egne testcase.

## 1.3 Omfang av dokumentet

Overordnet testplan beskriver omfang og gjennomføring av testing samt prosesser og rutiner som skal følges på tvers av de ulike testfasene.

Øvrige testplaner:

- Sprinttestplan pr. sprint
- Integrasjonstestplan

## 1.4 Definisjoner

Akseptansekriterier	Beskrivelse av krav til systemet som må være oppfylt for at fagsiden/systemeier skal akseptere systemet.
Akseptansetest	En formell test for å avgjøre om akseptansekriteriene er oppfylt og gi systemeier anledning til å akseptere systemet. Akseptansekriterier er definert ut fra krav innenfor omfang.
Avvik	Ethvert avvik mellom faktisk og forventet resultat, eller forhold som representerer et problem under testgjennomføringen dokumenteres som hendelser/ testobservasjoner. Nærmere undersøkelser kan føre til at avviket klassifiseres som feil (defekt) eller endringsanmodning. Se <i>Hendelse</i>
Endring	Endring i forhold til godkjente krav og design.
Enhet	En enhet er et sett med tjenester og/eller komponenter som til sammen utgjør en arbeidspakke for en utvikler.
Enhets-/ komponenttest	Test av det enkelte testobjekt som kan testes uavhengig av andre testobjekt eller av grupper av relaterte enheter. Formell test som utføres av utviklingsteamet.

	Se <i>Testobjekt</i>
Godkjenningskriterier	Beslutningsregler for å avgjøre om et testobjekt eller egenskap er godkjent i testen
Hendelse	Enhver hendelse som skjer under testutførelse som krever nærmere undersøkelse (IEEE 1008)
Ikke-funksjonell test	Ikke-funksjonelle tester er tester som i hovedsak har et teknisk utgangspunkt, for eksempel test av integrasjonsarkitektur, ytelse og lignende
Installasjonstest	En installasjonstest er en test for å verifisere om installasjonen lar seg gjøre vha. installasjonsdokumentasjon og at leveransen etter installasjon lar seg starte/kjøre.
Integrasjonstest	Test av samspillet mellom systemets deler (interne grensesnitt) og samspillet med andre definerte systemer som det skal virke sammen med (eksterne grensesnitt). Testing med hensikt å avdekke feil i måten testobjekt integrerer, med hensyn til å kalle hverandre korrekt, utføre komplette behandlingssekvenser og oversende og lagre data korrekt.
Kodedekning	En analysemetode for å måle hvilke deler av programvaren som har blitt utført og hvilke deler som ikke er blitt utført; for eksempel programinstruksjonsdekning, beslutningsdekning og forgreingsdekning (se <i>testdekning</i> ). Målingen krever at det benyttes verktøy.
Regresjonstest	Re-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 tilfredsstillende spesifiserte krav.
Retest	Gjentakelse av en test for å kontrollere at en feil er fjernet.
Risiko	Hendelse av negativ art som kan inntreffe
Smoketest	Et utvalg av alle definerte eller planlagte testtilfeller som dekker hovedfunksjonaliteten av testobjektet, for å sikre at de viktigste funksjonene i testobjektet virker, men uten å ta hensyn til mindre detaljer. Smoketest av daglig build av en ny release er del av beste industripraksis.
Stresstest	Stresstest er en test for å finne «knekkpunktet» til applikasjonen, dvs. hvilken maksimal belastning applikasjonen kan tåle før den begynner å feile.
Systemtest	Test av ferdig integrert system for å avdekke manglende samsvar med spesifiserte krav til systemet. Systemtest fokuserer på test av samspillet mellom rutiner og systemkomponenter og samspillet mellom systemet og konverterte data. Systemtest omfatter funksjonell test av systemet og ikke-funksjonelle tester som f.eks. stresstest, ytelsestest, usability test, osv.
Teknisk test	Teknisk test er test basert på både funksjonelle og tekniske testbetingelser, og som utføres av tekniske ressurser (eks. programmerere). Test av ikke-funksjonelle egenskaper som f.eks ytelse, svartider, stess.
Testbetingelse	Utdyper hva som skal testes og utledes av spesifikasjonen. Eks: hvis <betingelser>, så <hendelse/ tilstand>
Testcase/ testscript	En sekvens av sammenhengende handlinger med tilhørende kontroller relatert til testbetingelser. En detaljert beskrivelse av hvordan den enkelte del av testen skal gjennomføres.
Testdata	Oversikt over hvilke data som skal anvendes i et testscript. 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 retter seg mot alle spesifiserte krav for et gitt system eller komponent. For måling av testdekning benyttes ofte et verktøy.

Testgrunnlag	Den dokumentasjonen som testen blir basert på, for eksempel krav- eller designspesifikasjon.
Testmiljø	Testmiljø er det miljøet hvor testen skal foregå, både teknisk og fysisk. Det er vesentlig at dette miljøet er nært opp det fremtidige produksjonsmiljøet.
Testobjekt	Den minste testbare enhet. Størrelsen på enheten avhenger av hvilken testfase som gjennomføres. Se: <i>Enhet</i>
Testområde	Et funksjonelt område bestående av et eller flere testobjekter som skal testes sammen.
Testplan	Dokument som beskriver hensikt, omfang, fremgangsmåte, ressurser og fremdriftsplan for planlagte testaktiviteter innen utviklingsfasen. Videre defineres testobjekter, egenskaper som skal testes, testoppgaver og korresponderende ansvarsforhold, samt risikomomenter.
Testrapport	Dokument med en oppsummering av testaktiviteter og resultater. Inneholder også en evaluering av kvaliteten til de tilhørende testobjekter.
Teststrategi	Et dokument som beskriver testarbeidet (testprosessen) og de ansvarsforhold som gjelder i et prosjekt.
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.
Ytelsestest	Ytelsestest er en test for å måle responstider under gitte bruksmønstre i applikasjonen.

## 2 Overordnede prinsipper

### 2.1 Gradering av avvik

Ved rapportering av avvik i test skal det tas utgangspunkt i følgende definisjoner:

Nivå	Kategori	Beskrivelse
A	Kritisk feil	Feil som medfører at systemet stopper, at data går tapt eller at andre funksjoner som er kritiske for museene ikke er levert eller ikke virker som avtalt.
B	Alvorlig feil	Feil som fører til at funksjoner som er viktige for museene ikke virker som avtalt og som det er tids- og ressurskrevende å omgå.
C	Mindre alvorlig feil	Feil som fører til at enkeltfunksjoner ikke virker som avtalt, men som museene relativt lett kan omgå.
D	Andre avvik	Feil som er av kosmetisk art (skrivefeil, ikke pene skjermbilder, beregningsfeil på mindre viktige områder, steder man sjelden kommer innom, osv). Disse er som regel uproblematisk for videre testing og man kan også gå i produksjon med slike feil

### 2.2 Nettlesere

Det må som et minimum gjennomføres testing i følgende nettlesere:

- Microsoft Internet Explorer (versjon 10 og nyere)
- Google Chrome (nyere versjoner)
- Mozilla Firefox (nyere versjoner)

- Apple Safari (nyere versjoner)

## 2.3 Mobile plattformer

Det stilles i første omgang ikke krav til at all funksjonalitet skal fungere på mobile plattformer, men all informasjon må være tilgjengelig og mulig å fremvise. I tillegg vil det tilrettelegges for enkelte funksjoner.

Følgende plattformer må inngå i testen:

- Apple iOS (iphone, ipad)
- Google Android (mobil, tablet)

## 3 Gjennomføring av test

Dette kapitlet beskriver **hvordan** testingen skal gjennomføres.

Alle testscript prioriteres, følgende prioritering gjelder:

- 1-Høy
- 2-Medium
- 3-Lav

Under gjennomføring av test skal testscript behandles i prioritert rekkefølge på en slik måte at alle Script med prioritet 1-Høy utføres før prioritet 2-Medium, som igjen utføres før prioritet 3-Lav, så langt det er hensiktsmessig

Testleder vil i utgangspunktet fordele de forskjellige testscriptene til testerne.

### 3.1 Minimumstest

Det defineres ett sett med minimumstester som skal gjennomføres i hver test. Disse ligger i et eget testscript og oppdateres fortløpende.

### 3.2 Feilretting i testperioden

Rapporterte avvik rettes fortløpende og meldes tilbake Testleder for retest. Testleder bestemmer i samråd med utviklerteamet når ny deploy til testmiljø skal foregå. Deploy til testmiljø foretas av testleder.

### 3.3 Risikobasert testing

Alle tester er definert med utgangspunkt i kravspesifikasjoner og definerte brukerhistorier, og vil bli prioritert tilsvarende. Det vil i tillegg tas hensyn til omforent risikoanalyse.

### 3.4 Sprinttest

Detaljerte testplaner og testcase for neste sprint utarbeides parallelt med gjennomføring av foregående sprint. Test gjennomføres ved avslutning av hver sprint. Sprinttest gjennomføres i testmiljø.

- Testing av funksjoner utviklet i sprinten (funksjonelle krav, ikke-funksjonelle krav og eksterne grensesnitt)
- Regresjonstester av systemet etter hvert som det utvikler seg
- Konverterings- og migreringstester

Testingen er i hovedsak manuell testing som logges i Jira.

#### 3.4.1 Startkriterier

Før hver sprinttest:

- Detaljspesifikasjoner er beskrevet og godkjent i Jira
- Akseptanskriterier er etablert og godkjent av aktuell referansegruppe
- Det er ikke dokumentert A feil i utviklers enhetstest
- Utviklere har gjennomført enhetstest av utviklet funksjonalitet og dokumentert eventuelle avvik.
- Testleder mottar en liste over funksjonalitet som skal testes med spesifisering av eventuelle spesielle forutsetninger. Eventuelle avvik fra sprintplan spesifiseres.



- Testmiljø skal være klargjort for bruk. Testdata/migrerte data skal være lagt inn dersom det er påkrevd for testingen.
- Testleder verifiserer at eventuelle tilbakemeldinger fra tidligere sprinter er fulgt opp.
- Testscript er laget

### 3.4.2 Gjennomføring

Testene gjennomføres som angitt i testscriptene. Avvik føres inn i Jira og suppleres med skjermbilder og annen relevant dokumentasjon. Ferdig registrert avvik tilordnes testleder som sammenstiller og informerer prosjektgruppen i daglige «standup-møter» .

### 3.4.3 Utgangskriterier

- Alle planlagte testscript er gjennomført
- Avvik er dokumentert

## 3.5 Integrasjonstest

Integrasjonstest omfatter test av integrasjon mellom ulike microservices og systemer de har grensesnitt mot (se Grensesnitt pkt. 1.2 og Testmiljø pkt. 4)

## 3.6 Systemtest - Funksjonalitet

Systemtesten skal teste alle funksjonelle og tekniske sider ved systemet, og verifisere hvorvidt den utviklede leveransen inneholder avtalt funksjonalitet og kvalitet. Systemtesten inkluderer både faglige (funksjonelle) tester og ulike tekniske (ikke-funksjonelle) tester.

Hovedfokuset i systemtest er å avdekke så mange feil som mulig i den implementerte løsningen.

Det vil bli laget detaljerte testplaner for Systemtest pr. delleveranse.

## 3.7 Ytelsestest

Ytelsestest skal undersøke hvor vidt systemet tåler den belastning det vil utsettes for i drift og om respons og svartider tilfredsstiller oppdragsgivers krav. Ytelsestest tilrettelegges og testes ved hjelp av jMeter.

## 3.8 Stresstest

Hensikten med stresstest er å teste hvordan systemet oppfører seg når trafikkbelastningen er stor. Stresstest tilrettelegges og testes ved hjelp av jMeter.

Målsetningen med stresstesten er å måle den maksimale belastningen systemet tåler. Stresstest gjennomføres som del av Sprinttest.

## 4 Testmiljø

Testmiljø	Beskrivelse	Ansvar/ Eier
Utviklingsmiljø (U-miljø)	<b>Utviklingsmiljø.</b> I dette miljøet skal enhets/komponenttest gjennomføres. Test av integrasjon mellom enheter og løsningskomponenter skal gjennomføres så langt dette er mulig også under enhets-/komponenttestingen.	<b>Utviklingsteamet</b>
Test (T-Miljø)	<b>Testmiljø.</b> I dette miljøet gjennomføres sprinttest, integrasjonstest, systemtest og akseptansetest.	<b>Testleder</b>
Prod-miljø (P-miljø)	<b>Produksjonsmiljø.</b> Miljøet der den endelige løsningen gjøres tilgjengelig for sluttbrukerne.	<b>USIT</b>

## 5 Akseptanse og Godkjenning

Hovedfokus i en akseptansetest er å skape tillit til den utviklede løsningen. I akseptansetestfasen skal museene kontrollere at leveransen tilfredsstillende detaljspesifikasjonen og kravspesifikasjoner.

Akseptansetesten gjennomføres som en funksjonell test der de ulike prosessene testes i sammenheng. Akseptansetestene omfatter også test av robusthet (evnen til å tåle uforutsette hendelser som f. eks nettbrudd, input av urimelige data etc.). Testscript som benyttes i denne fasen vil derfor kunne avvike noe fra testscript benyttet under sprint- og systemtest.

Akseptansetesten gjennomføres av deltakere i referansegruppen for aktuell modul, samt representanter fra museene utpekt av Koordineringsgruppene.

Det utarbeides en testrapport fra akseptansetestfasen. Rapporten oppsummerer akseptansetesten og behandles av styringsgruppen. Med bakgrunn i innstilling i rapporten beslutter styringsgruppen om leveransen skal godkjennes eventuelt underkjennes.

Dersom styringsgruppen underkjenner leveransen, flyttes oppgaven tilbake til utviklerteamet som gjør ytterligere arbeid med funksjonen før den kan retestes og eventuelt godkjennes.

Det vil bli laget detaljerte planer for akseptansetest pr. delleveranse.