Stamdata

Testplan

Indholdsfortegnelse

[1 Formål 3](#_Toc304373506)

[2 Scope 4](#_Toc304373507)

[2.1 Serviceenablingen af autorisationsregisteret 4](#_Toc304373508)

[2.2 Kopi-register servicen 4](#_Toc304373509)

[2.3 Stamdata parsere 4](#_Toc304373510)

[3 Test strategi 5](#_Toc304373511)

[3.1 Unittest 5](#_Toc304373512)

[3.2 Integrationstest 5](#_Toc304373513)

[3.3 Performance- og stresstest 5](#_Toc304373514)

[3.4 Skaleringstest 5](#_Toc304373515)

[3.5 Endurancetest 5](#_Toc304373516)

[4 Hardwarekrav 7](#_Toc304373517)

[5 Miljøkrav 8](#_Toc304373518)

[5.1 Server 8](#_Toc304373519)

[5.2 Klient 8](#_Toc304373520)

[6 “Tidslinie” for testaktiviteter 9](#_Toc304373521)

[7 Testet funktionalitet / Acceptkriterier 10](#_Toc304373522)

[7.1 Serviceenablingen af autorisationsregisteret 10](#_Toc304373523)

[7.2 Kopi-register servicen 11](#_Toc304373524)

[7.3 Stamdata parsere 12](#_Toc304373525)

[8 Afhængigheder 13](#_Toc304373526)

[9 Værktøj 14](#_Toc304373527)

[10 Roller og ansvar 15](#_Toc304373528)

[11 Ændringslog 16](#_Toc304373529)

# Formål

I forbindelse med FMKi projektet, er der stillet krav om en række tests, der skal afvikles inden det endelige produkt kan blive deployet i produktion. Dette dokument indeholder en fuldstændig oversigt over de planlagte testaktiviteter, tidsplan derfor, og tilhørende accept-kriterier.

# Scope

De testaktiviter der beskrives i indeværende dokument, har til formål at kvalitetssikre de nævnte komponenter i forhold til det brugsmønster, der på skrivende tidspunkt forventes at være. Udgangspunktet for denne vurdering er primært baseret på FMKs forventede load mod systemet.

For alle komponenter gælder de generelle testkrav, at der skal være integrationstests og unittests med et code coverage på 80%. Derudover skal det sikres at de skrevne tests viser at den ønskede funktinoalitet er opfyldt, f.eks. ved at henvise til user-stories, der rammes af udførte tests.

Følgende komponenter og testaktiviteter er afdækket af dette dokument

## Serviceenablingen af autorisationsregisteret

Der er planlagt en simpel performancetest, der sikrer at kravspecs er overholdt for denne komponent.

## Kopi-register servicen

Der er planlagt en simpel performancetest, der sikrer at kravspecs er overholdt for denne komponent.

## Stamdata parsere

Der er planlagt en simpel performancetest af denne komponent, for at sikre at jobbet kan håndtere de forventede daglige datamængder.

# Test strategi

## Unittest

Som udgangspunkt bruges Emma og Cobertura til at måle code coverage, ud fra unittests lavet i JUnit frameworket. Målet er et minimums code coverage på 80%, der måles på antal instruktioner.

Der bør være 100% coverage på faktisk funktionalitet – med faktisk funktionalitet menes groft sagt sunshine scenarierne, og ikke fejlsituationer (e.g. exception-handling) eller lignende .

En Continious Integration server (Hudson) opstilles hos Trifork til at lave løbende rapporter over Code Coverage, samt at fange fejl i submitted code.

## Integrationstest

Værktøjet JUnit bruges til at definere og afvikle integrationstests.

Målet er at der er en 1-1 mapning mellem user stories i projektet og integrationstests. Et succesfuldt run på alle integrationstests bør kunne fortolkes sådan at alle user stories er opfyldt.

Det vil give god mening, under udviklingen af integrationstests, at løbende opdatere user stories, hvis man opdager scenarier det giver mening at integrationsteste (2-vejs kvalitetssikring).

## Performance- og stresstest

Der er stillet minimumskrav til performance for de enkelte services. Disse mål skal påvises opfyldt ud fra passende testcases. Så vidt muligt skal disse tests skrives så de kan genbruges, og med så lidt manuelt arbejde som muligt. JMeter baserede tests giver mening, da det primært er webservices der skal testes.

Testene skal kunne afgive læsbare rapporter, evt. understøttet af grafer, der kan påvise at den ønskede performance er understøttet.

## Skaleringstest

Skaleringstesten skal vise sammenhænget mellem svartider og stigened belastning indtil overbelastningspunktet nåes. Rapporten skal ligeledes indeholde oplysninger omkring spredning af svartiderne for hvert af målepunkterne.

## Endurancetest

Endurancetesten skal påvise at ingen af de udviklede komponenter degraderer over tid, dvs. svartider skal være “konstante”, og ressourcebrug på serveren skal stabilisere sig efter at komponenterne har kørt i en rum tid. Memory/Ressource leaks forventes fanget med denne test.

Endurance testen forventes afviklet lokalt i et nsp-in-a-box setup, hvor komponenterne for lov til at køre under medium load i en længere periode (en uges tid) – med løbende målinger på forbrugt hukommelse, svartider, diskplads, osv.

# Hardwarekrav

Performancetests skal endeligt afvikles på hardware der afspejler det faktiske produktionsmiljø. Den overordnede projektleder for FMKi (Jan Riis) har lovet at et præproduktionsmiljø bliver tilgængeligt, der har samme fysiske specs som produktionsmiljøet. Indledende performancetests vil blive afviklet på et lokalt udviklersetup, bestående af 3 Dell laptops med 4 GB ram og ~2 ghz dual-core cpu'er.

Til CI stilles ingen egentligt hardwarekrav, ud over at hardwaren selvfølgelig skal kunne afvikle alle unit- og integrationstests indenfor rimelig tid. I praksis er dette opfyldt ved den Hudson server som Trifork bruger til formålet.

# Miljøkrav

## Server

Server(ne) skal være konfigureret så de ligner det faktiske produktionsmiljø så meget som muligt. Til lokale udviklertests forventes det at vi kan bruge nsp-in-a-box til håndholdte tests.

Den endelige performance- og stresstest skal afvikles i et præproduktionsmiljø, der ligner produktionsmiljøet så meget som muligt (fx kunne det være passende at der kørte andre NSP komponenter under produktions-lignende load, mens testen blev afviklet).

## Klient

Da der ikke udvikles nogen faktiske klient komponenter, giver det ikke mening at stille krav til klienten, ud over at der skal være “nok” klienter tilrådighed, så vi kan sikre et fornuftigt load mod serveren. En række laptops, der sideløbende afvikler JMeter testcases forventes at være nødvendigt for at kunne sende det fornødne antal requests i sekundet.

# “Tidslinie” for testaktiviteter

|  |  |
| --- | --- |
| **Aktivitet** | **Udførselstidspunkt** |
| Udarbejdelse af testplan | Medio / ultimo marts |
| Udførsel af “udvikler” performancetests | Primo april – afventer dog tilgængeligheden af nsp-in-a-box, for at sikre en fornuftig testplatform |
| Udførsel af endurance test | Sker af 2 omgange – først en uge afsat til formålet når vi får adgang til nsp-in-a-box, og så en gentagelse af testen i forbindelse med selve leverancen, for at sikre at der ikke er introduceret nogle leaks i slutningen af projektet. |
| Udførsel af “produktions” performancetests | Primo maj – afventer dog tilgængeligheden af et præproduktionsmiljø |
| Udvikling af unittests | Sker sideløbende med udviklingen af funktionaliteten |
| Udvikling af integrationstests | Sker sideløbende med udviklingen af funktionaliteten |

# Testet funktionalitet / Acceptkriterier

Som udgangspunkt testes alt nævnt i afsnit 2 (Scope) af dette dokument. Dette afsnit går lidt mere i dybden med hvad præcist der testes, og under hvilke forhold testen siges at have været en succes.

Performancekravet for alle udstillede services siger at servicen skal kunne håndtere 10 samtidig kald, og svartiderne skal være i overensstemmelse med normal praksis indenfor området.

## Serviceenablingen af autorisationsregisteret

Autorisationsregisterservicen vil blive udsat for følgende tests, der dækker kravet om performance- og stresstest, endurancetest og skaleringstest. For hvert test-type dokumenteres gennemsnitlig svartid, samt spredning af svartider. Det dokumenteres også hvis servicen begynder at give fejl (eg http 500).

**Performance- og stresstest samt skaleringstests.**

En JMeter klient laver stigende load mod serveren, hvor de ovennævnte målepunkter noteres.

* 1 req/sek (forventet brugsmønster)
* 2 req/sek
* 5 req/sek
* 10 req/sek (peak brugsmønster)
* 50 req/sek
* 100 req/sek

**Endurancetest**

En JMeter klient laver kontinuerlige requests mod serveren, et enkelt request pr sekund, over en uge, hvor der dagligt afmåles følgende oplysninger fra serveren

* svartider på servicen (gennemsnit og fordeling over en periode på 10 minutter)
* serverens hukommelsesforbrug
* serverens diskforbrug
* serverens CPU belastning

Acceptable svartider for servicen under peak brugsmønster (10 req/sek) er følgende:

* gennemsnitlig svartid på < 750 ms
* 95% af alle kald skal være < 1000 ms
* 99% af alle kald skal være < 2000 ms

Ligeledes er det et acceptkriterium at hukommelsesforbrug og CPU-belastning stabiliseres indenfor en rum tid efter at serveren er startet op.

## Kopi-register servicen

Kopiregisterservicen vil blive udsat for følgende tests, der dækker kravet om performance- og stresstest, endurancetest og skaleringstest. For hvert test-type dokumenteres gennemsnitlig svartid, samt spredning af svartider. Det dokumenteres også hvis servicen begynder at give fejl (eg http 500).

For alle tests gælder det at målingen er på det initielle kald til webservicen, og ikke på de efterfølgende dataudtræks kald. Der laves sideløbende tests på throughput af dataudtrækket.

**Performance- og stresstest samt skaleringstests.**

En JMeter klient laver stigende load mod serveren, hvor de ovvennævnte målepunkter noteres.

* 1 req/sek (forventet brugsmønster)
* 2 req/sek
* 5 req/sek
* 10 req/sek (peak brugsmønster)
* 50 req/sek
* 100 req/sek

**Endurancetest**

En JMeter klient laver kontinuerlige requests mod serveren, et enkelt request pr sekund, over en uge, hvor der dagligt afmåles følgende oplysninger fra serveren

* svartider på servicen (gennemsnit og fordeling over en periode på 10 minutter)
* serverens hukommelsesforbrug
* serverens diskforbrug
* serverens cpu belastning

Acceptable svartider for servicen under peak brugsmønster (10 req/sek) er følgende

* gennemsnitlig svartid på < 750 ms
* 95% af alle kald skal være < 1000 ms
* 99% af alle kald skal være < 2000 ms

Ligeledes er det et acceptkriterium at hukommelsesforbrug og cpu belastning stabiliseres indenfor en rum tid efter at serveren er startet op.

Som nævnt udføres der en ekstra test på throughput af data gennem dataudtræksdelen af kopi-register-servicen. Der laves throughput målinger (kb/sec) på følgende scenarier

* 1 klient, der kopierer et helt register
* 2 selvstændige klienter (hver deres PC), der kopierer et helt register
* 3 selvstændige klienter (hver deres PC), der kopierer et helt register
* 10 “simulerede” klienter (2 PC'ere med 5 tråde hver), der kopierer et helt register

Der stilles ingen direkte krav til acceptabelt throughput, men der laves en vurdering af resultatet af testen, i forhold til størrelsen af de enkelte registre, og antallet af brugere af kopi-register-servicen (fordelt på NSP instanser)

## Stamdata parsere

Der er planlagt en simpel performancetest af denne komponent, for at sikre at jobbet kan håndtere de forventede daglige datamængder.

# Afhængigheder

* Adgang til et præproduktionsmiljø hvor performancetests kan afvikles
* Adgang til et nsp-in-a-box image, hvor udvikler-performancetests kan afvikles

# Værktøj

JMeter, Emma, Hudson, Cobertura – måske nogle rapporteringsværktøjer

# Roller og ansvar

Brian Graversen, projektleder FMKi spor 6

* Sikre at de beskrevne tests bliver afviklet i de angivne miljøer.
* Være med til at sikre at tidsplanen bliver overholdt.

Anders Sørensen, ?

* Kvalitetssikre testplanen og afviklingen af testaktiviteterne

Christian Ernstsen, ?

* Være med til at sikre at tidsplanen blive overholdt.
* Sikre at afviklingsmiljøerne er tilgængelige hos operatøren/driften

# Ændringslog

Kilden til dette dokument kan findes på:

<https://svn.softwareborsen.dk/stamdata/trunk/Dokumentation>

| Version | Dato | Ændring | Ansvarlig |
| --- | --- | --- | --- |
| 1.0 | 2011-04-27 | Initielt dokument | bbg@trifork.com |