Frederik Bothmann Larsen

Test Essay

The 4 agile testing quadrants

The 4 agile testing quadrants er en model som kategorisere forskellige typer test. De kategorier er delt op i kvadranter og 2 af kvadranterne kan falde i samme kategori. F.eks. har vi på venstre side 2 kvadranter som udgøre tests der skal kunne supporte selve udviklings holdet, her ligger tests som unit test og functional tests. I den anden side har vi tests der skal kunne kritisere produktet, her ligger performance tests og exploratory test, denne side skal kunne teste hvor godt produktet nu gør det den skal kunne gøre. Hvor den anden side er til at teste hvorvidt programmet gør det den skal kunne.

De 2 øverste kvadranter udgøre business/customer facing. Her ligger tests som simulationer og usability test. Det er her vi har de tests som oftes involvere eller bliver krævet af customers.
Nedenunder de 2 kvadranter har vi de Technology facing kvadranter. Det er de tests vi laver som customeren ikke har noget at gøre med, som reelt er det bare tests som tester at teknologien gør hvad den skal.

System testing

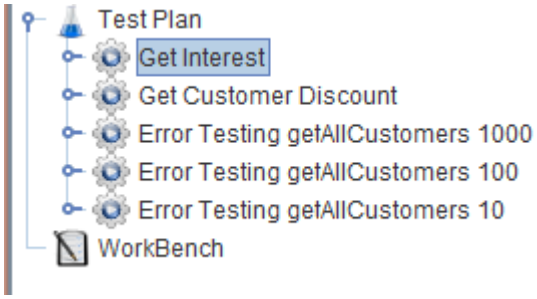
System testing er niveauet før acceptance testing, det er her vi tester systemets helhed, og sikre os at programmet/softwaren møder kravene. Testene bliver oftes lavet udfra bl.a. risks eller usecases. Der bliver både testet functional og non-functional tests i system testing, da system testing indebære alt.

Exploratory testing

Exploratory testing er en slags blackbox testing, da idéen med det er at testeren skal lære softwaren at kende mens han/hun tester det. Mens testen bliver udført videre bygger man på test designet eller finder på nye designs, med andre ord bliver design og udførelsen gjort parallelt. I modsætning til andre blackbox teknikker er der ikke meget dokumentation før testen, eller ikke meget planlægning. Men der er derimod meget dokumentation undervejs i processen. Der bliver noteret ned hvad der bliver gjort og hvad resultatet af det er. En slags log.

Frederik Bothmann Larsen

JMeter summary



Testplanen består af 5 test groups.

1. Get Interest, tester getinterest med en csv der ser således ud: Csv'en er bygget efter krænseværdier.

```
-1,0
0,0
1,3
99,3
100,3
101,5
999,5
1000,7
1001,7
```

2. get Customer Discount tester getCustomerDiscount med en csv der ser således ud:

```
true,true,false,0
true,false,true,20
true,false,false,15
false,true,true,30
false,true,false,10
false,false,true,20
false,false,false,0
```

3. Error Testing getAllCustomers 1000, tester om serveren kan klare 1000 calls lige efter hinanden

4. Error Testing getAllCustomers 100, tester om serveren kan klare 1000 calls lige efter hinanden

5. Error Testing getAllCustomers 10, tester om serveren kan klare 1000 calls lige efter hinanden

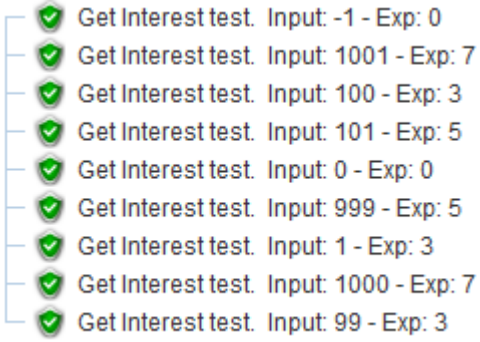
Ganied knowlegde

Udfra disse tests, sikre vi os at serveren udnytter metoderne rigtigt og at den håndtere inputs og outputs korrekt. Vi ved også at serveren kan håndtere mange kald på en gang.

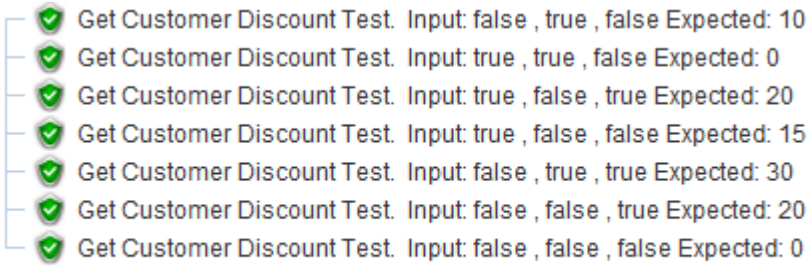
Frederik Bothmann Larsen

Resultater

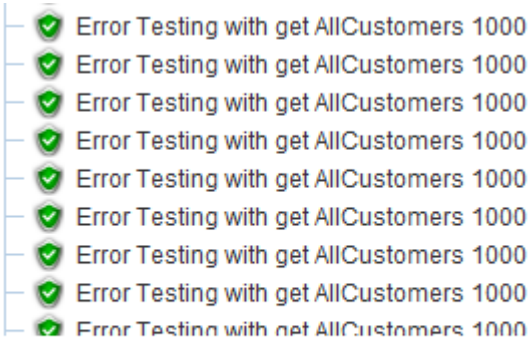
1. Get Interest



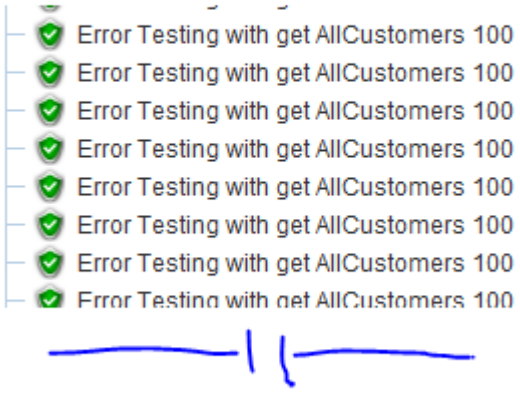
2. Get Customer Discount



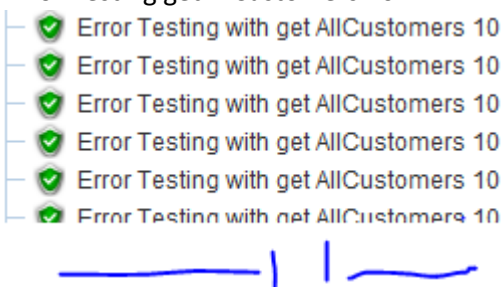
3. Error Testing getAllCustomers 1000



4. Error Testing getAllCustomers 100



5. Error Testing getAllCustomers 10



Sådan ser hele test planen ud: