

# System testing

Nikolai Hansen

## The 4 agile testing quadrants

### Quadrant 1

Unit test er hvor man tester funktionalitet for en komponent eller unit, af systemet. Det vil sige små isoleret test af funktionalitet af et system som fx en method. Når der er lavet unit test og de er bestået kan man dokumenter adfærd i systemet, og være sikker på at den del af koden der bliver testet virker efter hensigten.

Component tester større dele af systemet.

### Quadrant 2

Test i quadrant 2 minder om test i quadrant 1. I quadrant 1 laver man customer test, det vil sige at man finder ud af hvad der skal testes ud fra customer eksempler. De bliver skrevet på en måde så 'business experts' let kan læse og forstå dem. På den kan 'business experts' vurdere om systemet er lavet efter hensigten.

Det overordnede formål med quadrant 1 og 2 er at lave nogle hurtig test, så man hurtig kan finde evt. fejl og bedre kunne udvikle kode.

### Quadrant 3

I quadrant 3 laver man 'Business case' test så som test af scenarios. Et scenario kan være en rutine en customer har i virksomheden som skal implanteres i systemet. Det bliver brugt til at validere hvordan en rigtig user vil bruge systemet.

### Quadrant 4

I Quadrant 4 tester man non-fuctional krav til systemet (FURPS). Det kunne fx være 'Performance', hvis det var tale om udvikling af en web side kunne dette være respons tid på web siden. Det kunne også være sikkerheds krav til system fx alle password skal krypteres.

## System testing

System testing er testing af en komplet program. Formålet med system testing er at tjekke om kravene bliver overholdt. Man laver functional eller non-functional test. I vores eksempel med discount kunne det være at test at når brugeren har en kupon så skal brugerne have 20% rabat.

## Exploratory testing

Er en bestemt måde at tænke på, ikke en testing teknologi. Exploratory testing går ud på at lære at om den testede software, samtidig med at lave nye test. Fordelen ved exploratory testing, er at det ikke kræver meget forberedelse, og at man finder fejl hurtigt.

## Jmeter

### Functional tests

**Interest** – parameterized med csv fil

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	22:12:23.411	Interest_Test 1-1	HTTP Request	149	✓	312	218	149	36
2	22:12:23.511	Interest_Test 1-2	HTTP Request	111	✓	314	218	111	34
3	22:12:23.610	Interest_Test 1-3	HTTP Request	98	✓	314	219	98	33
4	22:12:23.710	Interest_Test 1-4	HTTP Request	110	✓	311	219	110	38
5	22:12:23.811	Interest_Test 1-5	HTTP Request	109	✓	336	219	109	37
6	22:12:23.911	Interest_Test 1-6	HTTP Request	108	✓	312	220	108	34
7	22:12:24.010	Interest_Test 1-7	HTTP Request	107	✓	315	220	107	34
8	22:12:24.112	Interest_Test 1-8	HTTP Request	112	✓	312	220	112	43
9	22:12:24.211	Interest_Test 1-9	HTTP Request	103	✓	312	218	103	36
10	22:12:24.310	Interest_Test 1-10	HTTP Request	111	✓	314	218	111	36

Text

- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request

Sampler result

Request

Response data

GET http://cybermanager.dk/systemTesting/index.php?balance=99

GET data:

[no cookies]

Request Headers:

Connection: keep-alive

Content-Length: 0

Content-Type: text/plain; charset=ISO-8859-1

Host: cybermanager.dk

User-Agent: Apache-HttpClient/4.5.2 (Java/1.8.0\_121)

**Discount** - parameterized med csv fil

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	23:34:43.955	Discount_Test 1-2	HTTP Request	193	✓	311	254	193	59
2	23:34:44.055	Discount_Test 1-3	HTTP Request	99	✓	310	254	99	34
3	23:34:44.154	Discount_Test 1-4	HTTP Request	103	✓	310	253	103	36
4	23:34:44.254	Discount_Test 1-5	HTTP Request	106	✓	330	256	106	36
5	23:34:44.355	Discount_Test 1-6	HTTP Request	101	✓	311	255	101	33
6	23:34:44.455	Discount_Test 1-7	HTTP Request	105	✓	311	255	105	37
7	23:34:44.555	Discount_Test 1-8	HTTP Request	104	✓	311	254	104	36
8	23:34:44.755	Discount_Test 1-9	HTTP Request	104	✓	311	254	104	34
9	23:34:43.855	Discount_Test 1-1	HTTP Request	1232	✓	311	255	1232	163
10	23:34:44.656	Discount_Test 1-9	HTTP Request	433	✓	311	255	433	33

Text

- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request

Sampler result

Request

Response data

GET http://cybermanager.dk/systemTesting/index.php?newCustomer=true&loyaltyCard=true&coupon=false

GET data:

[no cookies]

Request Headers:

Connection: keep-alive

Content-Length: 0

Content-Type: text/plain; charset=ISO-8859-1

Host: cybermanager.dk

User-Agent: Apache-HttpClient/4.5.2 (Java/1.8.0\_121)

## Non-functional tests

### Aggregate report - Discount

**Aggregate Report**

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only: ☐ Errors ☐ Successes 

Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	10	105	99	109	109	158	94	158	0,00%	2,2/sec	0,66	0,54
TOTAL	10	105	99	109	109	158	94	158	0,00%	2,2/sec	0,66	0,54

### Summary report - Interest

**Summary Report**

Name: Summary Report

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only: ☐ Errors ☐ Successes 

Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	10	100	95	115	5,80	0,00%	10,0/sec	3,10	2,14	317,3
TOTAL	10	100	95	115	5,80	0,00%	10,0/sec	3,10	2,14	317,3

### Throughput time - interest

Users	Sec
10	1,1
100	9,9
1000	98,0

Jmeter er en god måde at lave customer test på et system. I mine functional test lavet jeg parameterized test med csv, på den måde kan jeg let test flere input og de forventet ouput, og tjekke at der kommer en 200 response code tilbage.

I de non- functional test har jeg lavet en 'aggregate report', hvor man kan se Min, Max, Average, Error og Throughput, på den måde kan jeg se om sytemet overholder evt. non-functional krav, fx hvis man har et non-functional krav at en service ikke må tage over et bestemt tidsinterval.