

Teknisk testning för otekniska testare

SAST, 16-feb-2017

Rikard Edgren

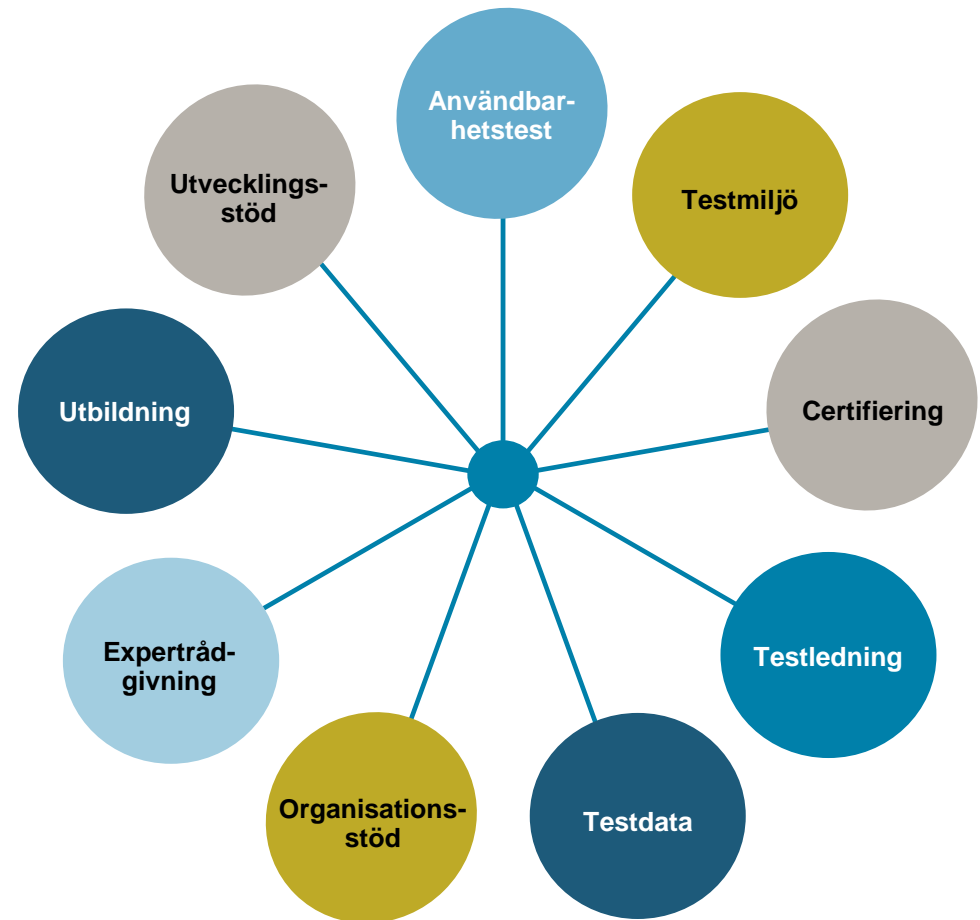
Nordic Medtest

rikard.edgren@nordicmedtest.se

Nordic Medtest utför testning och kvalitetssäkring och bidrar till mer användbar och säkrare IT i vården.

Vi är:

- En offentlig verksamhet
- Oberoende och utan vinstintressen
- Proffs på nationell e-hälsa



Vad är teknisk testning?

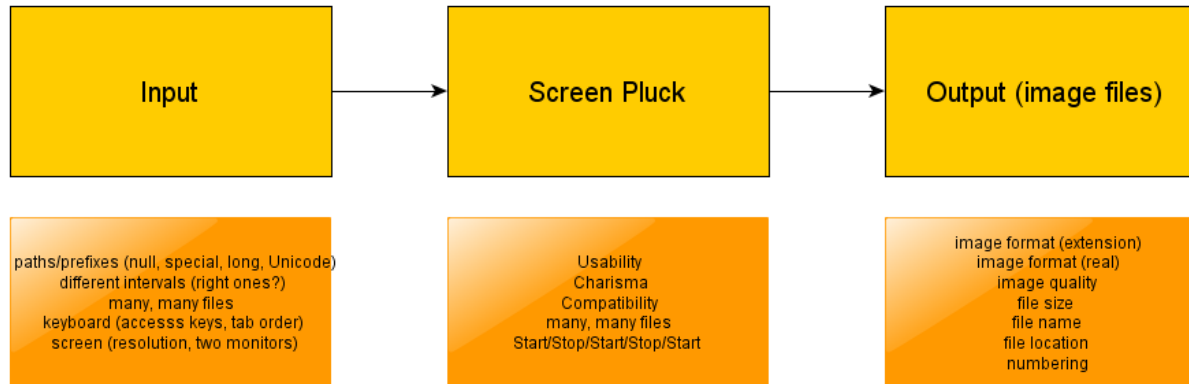
▶ ???

- Att man kan programmera?
 - Att man använder och förstår avancerade verktyg?
 - Att man automatiserar?
 - Att man tittar under huven?
 - Att man pratar samma språk som utvecklarna?
 - Det som görs av tekniska testare?
 - Att man tänker på ett tekniskt sätt?
- ▶ *att använda kunskap om den teknik som används, och som kan användas, för att designa och utföra tester, inklusive att observera testobjektet.*

9 Tips om teknisk testning

1. Det generella systemet är arkitektur
2. Tekniskt språk
3. Utvecklarverktyg
4. Läsa (och förstå?) loggfiler
5. Skriptade testdata
6. Automatiseringens syfte och omfång
7. Testverktyg
8. Be om hjälp
9. Glöm inte det otekniska

Det generella systemet är arkitektur



- ▶ Om du förstår det generella systemet kan du prata arkitektur
 - Vilka övergripande komponenter finns?
 - Vem pratar med vem?
 - Vad är input och output?
- ▶ Samt förstås kritiskt granska detta och få massvis med testidéer.

Tekniskt språk

- ▶ Alla behöver inte kunna programmera.
- ▶ Men kommunikationen kommer bli mycket bättre om du förstår och kan använda samma språk som utvecklarna.
- ▶ Du får också mer respekt, vilket alla har nytta av.



Loggfiler

- ▶ Alla applikationer har loggfiler som det finns intressanta saker i.
- ▶ Alla applikationer har loggfiler som det finns massvis av ointressanta saker i.
- ▶ En bra start (efter att få tillgång till loggfilerna!) är att söka efter saker som inte borde finnas där (ERROR, Exception m.m.)
- ▶ När du väl är inne i loggfilerna, fundera på om något mer borde loggas, för att öka applikationens testbarhet.

Skriptta testdata

- ▶ Ofta blir inte testningen bättre än den testdata man har.
- ▶ Det är lättare än du tror att skriptta fram rikare och mer testdata.

```
23 while $i < 1001 do
24   # Skapa X antal noder
25   $date = "201#{1 + rand(9)}0#{1 + rand(9)}000000"
26
27   tmpa = "a#{1 + rand(9)}"
28   tmpb = "b#{1 + rand(9)}"
29   diagnoskod = worksheet.Range(tmpa).Value # Get diagnosis code from cell A#i in Excel sheet
30   diagnostext = worksheet.Range(tmpb).Value # Get diagnosis name from cell B#i in Excel sheet
31
32   output_file.puts("      <urn:diagnosis><urnl:diagnosisHeader>
33     <urnl:documentId>JOL-MOCK-GD-04-#{1 + rand(9)}</urnl:documentId>
34     <urnl:sourceSystemHSAId>JOL-MOCK</urnl:sourceSystemHSAId>
35     <urnl:documentTime>#{1 + rand(9)}000000</urnl:documentTime>
36     <urnl:patientId>
37       <urnl:id>189908029805</urnl:id>
38       <urnl:type>1.2.752.129.2.1.3.1</urnl:type>
39     </urnl:patientId>
40     <urnl:accountableHealthcareProfessional>
41       <urnl:authorTime>#{1 + rand(9)}000000</urnl:authorTime>
42       <urnl:healthcareProfessionalOrgUnit>
43         <urnl:orgUnitHSAId>healthcareProfessionalOrgUnit</urnl:orgUnitHSAId>
44         <urnl:orgUnitName>orgUnitName</urnl:orgUnitName>
45       </urnl:healthcareProfessionalOrgUnit>
46       <urnl:healthcareProfessionalCareUnitHSAId>TSTNMT2321000156-1003</urnl:healthcareProfessionalCareUnitHSAId>
47       <urnl:healthcareProfessionalCareGiverHSAId>TSTNMT2321000156-1002</urnl:healthcareProfessionalCareGiverHSAId>
48     </urnl:accountableHealthcareProfessional>
49     <urnl:approvedForPatient>true</urnl:approvedForPatient>
50   </urnl:diagnosisHeader>
51   <urnl:diagnosisBody>
52     <urnl:typeOfDiagnosis>Huvuddiagnos</urnl:typeOfDiagnosis>
53     <urnl:diagnosisTime>20151201080000</urnl:diagnosisTime>
54     <urnl:chronicDiagnosis>true</urnl:chronicDiagnosis>
55     <urnl:diagnosisTime>20151201080000</urnl:diagnosisTime>
56     <urnl:diagnosisCode>
57       <urnl:code>#{1 + rand(9)}</urnl:code>
58       <urnl:codeSystem>1.2.752.116.1.1.1.3</urnl:codeSystem>
59       <urnl:displayName>#{1 + rand(9)} post #{1 + rand(9)}</urnl:displayName>
60     </urnl:diagnosisCode>
61   </urnl:diagnosisBody>
62   </urn:diagnosis>")
63   $i=$i+1
64 end
```

Automatiseringens syfte och omfattning

- ▶ Om du är en ”oteknisk” testare, så finns det andra som är oerhört mycket bättre på att skapa tester som körs av datorer.
- ▶ Men har du god kunskap om produkten och dess användning, så kanske du är mycket bättre på att bestämma vad som ska testas av datorer, och varför.
- ▶ Samarbete är inne!
- ▶ Hört talas om semi-automatiserad testning?

Testverktyg

- ▶ Det är coolt med mjukvara, och det finns både komplicerade och enkla program (och flera som är både och).
- ▶ Här är några tekniska gratisverktyg, som vem som helst kan köra:
 - **Fiddler:** Starta, så får du mer information om nätverkstrafiken.
 - **Databasverktyg:** Bara att installera, få lösenord, titta i tabellerna
 - **Selenium IDE:** Spela in automatiska webb-tester med Firefox, lägg på "verify" med högerklick; det blir inget robust och långsiktigt, men kan snabbt visa vad man kan testa.
 - **Zed Attack Proxy:** Installera, sätt Local Proxy 8080, och 8080 som proxy i din webbläsare, kör applikationen och titta på alerts från ZAP.
 - **JMeter:** Kopiera till din dator, hitta PDF:en som beskriver Test Script Recorder, spela in något, och lek.

WARNING: Kör inte prestanda- och säkerhetstestverktyg på sajter utan tillstånd!

Be om hjälp

- ▶ Man kan lära sig mycket, men inte allt.
- ▶ Du kommer långt om du lär dig tillräckligt mycket om tekniken så att du vet vad som kan göras, och sen ber andra om hjälp för att göra det.
- ▶ Förstå tekniken kontra att bemästra den.

Glöm inte det otekniska

- ▶ I all teknik får man inte tappa bort det humanistiska.
- ▶ Funkar för en maskin är inte samma som att det funkar för människor.
- ▶ Du får titta på lampan, om du också tittar på produkten, ur intressenternas perspektiv.



Exempel: SoapUI

The screenshot shows the SoapUI 5.3.0 interface. The main window displays a test suite named "4 Content tests - DirectAccess". Underneath, there is a test case "4.1 ApprovedForPatient_true" which contains a test step "Call GetMedicationHistory". The "Inspector" panel on the right shows the XML request and response for this test step. The request is a SOAP envelope with a header containing a logical address and a body containing a "GetMedicationHistory" request with patient ID and type parameters. The response is a SOAP envelope with a header and a body containing a "GetMedicationHistoryResponse" with a "MedicationMedicalRecord" containing various details like document ID, source system, patient ID, and healthcare professional information.

Custom Properties

TestRequest Properties	
Property	Value
Name	Call GetMedication...
Description	
Message Size	674
Encoding	UTF-8

Properties

SoapUI log http log jetty log error log wsrm log memory log SoapUI

- ▶ Scheman och Schematron kontrollerar att alla regler följs.
- ▶ Människor bedömer om **innehållet** är rätt.

Sammanfattning

- ▶ Testning är svårt, tacka gudarna för det!
- ▶ Det handlar om både teknik och människor, ingen kan allt, men man missar en del om man ignorerar ena delen helt.
- ▶ Om du breddar dina perspektiv, så kan du bidra med mer värde.

Teknisk testning för **humanistiska** testare

SAST, 13-feb-2017

Rikard Edgren

Nordic Medtest

rikard.edgren@nordicmedtest.se