

System – och användarkrav resultathantering

TUC

Onsdag 27/4

<https://github.com/Rnqst>

Planering

27/4 Onsdag – Kravanalys / specifikation, resultathantering

28/4 Torsdag – Analys av testresultat, avvikelser

4/5 Onsdag – Olika alternativ för hantering av resultat

5/5 Torsdag – Hemtentamen, möjlighet till frågor

Dagsschema

Tider:

9:00 - 9:45 Lektion
10:00 - 10:45 Lektion
11:00 - 11:45 Lektion
12:30 - 13:15 Lektion
13:30 - 14:15 Laboration
14:30 - 15:30 Laboration
15:30 - 16:00 Sammanfattning

Introduktion – kravanalys och specifikation

- Användbarhetsfrågor
- Användarkrav
- Verksamhetsmål, fokus och filosofi

Mjukvarukvalitet enligt ISO-9126

- Funktionalitet - förmåga att uppfylla de funktionskrav som ställs
- Tillförlitlighet – att fungera när det behövs
- Användbarhet – intuitiv, smidig, leverera resultat (mervärde)
- Effektivitet – utifrån fysiska och mänskliga resurser
- Underhållbarhet – är koden läsbar? "vettigt" programmeringsspråk? Vettiga beroenden?
- Portabilitet – går den att flytta till ett annat operativsystem?

Funktionalitet

- Användarfall (use cases)
- Agil utveckling
- Återkoppling från support och sälj

Tillförlitlighet

- Rimliga beroenden och begränsad komplexitet
- Försiktig och enkel design (statisk analys, kodrevision)
- "vanlig" funktionstest
- Stresstest, dvs test under svåra förhållanden, lite minne, hög last m.m.
- Test med simulerade fel i driftsmiljön

Användbarhet

- Starka produktägare, program management
- QA med pondus
- Support, kunder
- Test hos kund

Effektivitet

- Fysisk effektivitet
- Systemkrav
- Systemkomplexitet
- Databaser
- Supportprogram
- Virtualisering
- Moln

Effektivitet

- Mänsklig effektivitet
- Utbildningskrav
- Intuitiva funktioner
- Dolda funktioner
- Smidiga arbetsflöden för vanliga uppgifter
- Svarstider

Underhållbarhet

- Enkel är ofta smidig
- Få beroenden
- Läsbar kod
- Omsorgsfulla val av verktyg och programmeringsspråk
- Väl valda underleverantörer

Portabilitet

- Omsorgsfull design
- Vettigt programmeringsspråk
- Väl valda beroenden och komponenter
- Testa på flera plattformar

Har vi fler standarder?

- När används standarder?
- Vad är syftet med standarder?
- Hur påverkar det vårt arbete inom QA?

Testfallsresultat

- Tydliga eller entydiga resultat
- Halvtydliga resultat
- Bilder och liknande
- Rätt resultat men fel ändå
- Kanske rätt resultat?
- Numeriska resultat och liknande

Tydliga eller entydliga resultat

- Bra!
- Eller inte?
- Användbarhetsbrister
- Säkerhetsbrister
- Tillförlitlighet
- Orimliga svarstider

Halvttydliga resultat

Exempel, du söker på "kran" för att få hjälp med din vattenkran och hamnar på en sida med lyftkranar..

- Definiera acceptabla och oacceptabla resultat beroende på vad som är mest praktiskt
- Försök ringa in otydligheterna
- Gränsfallen alltid intressanta

Bilder och liknande

Testresultat där bilder och annan media skall utvärderas är alltid besvärlig.

- Försök hitta tydliga fall
- Grovsortera med verktyg
- Utnyttja nättjänster, exempelvis googles API

Rätt resultat, men ändå inte

Gränsfallen alltid besvärliga. Användbarhetsfrågor får inte glömmas bort.

- Sök vägar att grovsortera
- Sök testfall som ger tydliga resultat

Numeriska resultat

Textsträngar, symboler etc.

- Gränsfall

Dels för vad som är acceptabelt ur ett systemperspektiv, men även för testanalys.

- Statistisk behandling

Testfallsresultat

- Tydliga eller entydiga resultat
- Halvtydliga resultat
- Bilder och liknande
- Rätt resultat men fel ändå
- Kanske rätt resultat
- Numeriska resultat och liknande

Resultatanalys är inte alltid enkel

Involvera kollegorna

- Förankra dina egna iakttagelser och ditt sätt att validera resultat
- Är dina gränsvärden bra?
- Har du vettiga intervall?
- Känner någon annan till viktiga värden eller utfall?

Testdata

Förankra ditt testupplägg rent fysisk, vad tycker dina kollegor? Har du rätt ingångsvärden?

- Testar du med rätt konfiguration?
- Använder du bra ingångsvärden (exempelvis den databas som ditt program använder)?
- Stressar du på rätt saker? Är det likvärdigt att använda en äldre dator med sämre CPU och mindre minnesmängd och då köra på en lägre lastnivå jämfört med de moderna högprestanda maskiner som dina kunder använder för att köra på betydligt högre belastning?

Feltillstånd och yttre problem

- Vilka feltillstånd skall vårt system klara av och hur testar vi detta?
- Vad förväntas hända med t.ex. en ny faktura om operatören matat in den, men nätsladden rycks ur precis innan denne tryckt på "enter"? Skall det sparas när nätsladden kopplas in igen?
- Vad händer vid strömavbrott?

Här måste vi både tänka efter själva och ta hjälp så att vi vet att vi tänkt rätt och så att vi testar den här sortens gränsfall på ett vettigt sätt.

En realistisk testmiljö

- Test av installationsrutiner är viktigt, men det får inte ta tid från andra saker
- Testdata som liknar verkligheten
- Hårdvara, datorer, nätverk, uppkoppling som liknar era användares miljö
- Stresstestar vi

Testdata

- Realistiska ingångsvärden
- Fyllda databaser
- Se till att ha med några "intressanta" extremvärden

Hårdvarumiljö

- Sitter dina användare på dåligt mobilnät, måste ni åtminstone kunna simulera ett sådant
- Har brukarna tio år gamla maskiner måste ni också köra antika maskiner
- Där det inte går att efterlikna användarnas miljö i detalj, improvisera, simulera...
- Lasta maskinerna tills de går trögt ifall ni inte får tag i gamla maskiner, samma med nätet med en modern switch eller router begränsar du lätt trafikflödet

Stresstest

- Relevanta stresstester är krävande att sätta upp
- Få med så många i organisationen du kan för att utvärdera ditt testupplägg
- Belastning är inte linjär, detta gör att dina lastfall måste vara genomtänkta