

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

Cem Kaner —“The best test cases are the ones that  
find bugs.”

# Introduktion – testresultat och avvikelser

- Hitta och förstå fel
- Logga avvikelser, prioritera och hantera dem
- Mäta testprocessen, hur långt har vi kommit...

# Hitta och förstå fel

- Hur kommunicerar vi med utvecklarna, närhet...
- Ringa in fel, vad fungerar? Vad kraschar?
- Kategorisera felets grad

# Hitta och förstå fel

- Är vi del av felsökningen eller är vi bara försökskaniner som rapporterar när det strular? Är vi del i utvecklingen eller är vi bromsklossar?
- Regressioner kan vara tecken på bakomliggande designfel

# Hitta och förstå fel

- QA har nästan alltid bättre testmiljö än utvecklarnas
- QA är inte alltid en aktiv del när felen skall rättas till

# Logga avvikelser, prioritera och hantera dem

- Bra felrapporter med relevant information; svårt..
- Du måste själv ha en uppfattning om hur svårt ett fel är
- Var aktiv, följ upp, kontakta chefer, utvecklare...



# Mäta testprocessen

Hur går det? Hur långt har vi kommit? Tror vi att vi börjar bli klara?

Även vi aldrig kan vara säkra är testprocessen vårt enda redskap att mäta hur applikationen "mår" och hur mycket mer arbete som krävs innan leverans.

# Hur kommunicerar vi med utvecklarna

Om vi kan (får) kontakta utvecklarna direkt så kan vi ofta bidra till deras felsökningsarbete i koden även om vi själva inte alltid förstår hur koden fungerar. Vi har bättre testresurser och vi måste få utvecklarna att förstå detta.

Då blir du även en del i deras lag.

# Ringa in fel

- Förbättra testfall
- Förbättra testmiljön
- Hitta kritiska indata som utlöser buggen (och hitta data som inte gör det...)
- Sammarbeta med utvecklarna

# Kategorisera felets grad

- Vi är användarnas ombud i utvecklingsarbetet, därför måste vi ha en åsikt
- Projektledare, program management, utvecklarna och de andra intressenterna har sina agendor
- Tekniska kriterier

# Tekniska kriterier för bugklassificering

- Hindrar det från att använda produkten eller hindrar det någon central funktion?
- Finns det något alternativt sätt att komma runt problemet?
- Är buggen bara irriterande, eller hindrar den någonting viktigt?

# Förbättra testfall

Utvecklarna är duktigare på koden, vi är bättre på att testa; låt testarna vara en aktiv del i felsökning och utveckling.

Ta initiativ själv, du får roliga arbetsuppgifter och bättre anställningstrygghet, dessutom blir du inte bara en i gänget i testlaget utan du "infiltrerar" utvecklarteamet.

Förbättra testfall, ett testresultat leda till:

- Förbättra testfallet som "hittade" buggen
- Skapa nya testfall
- Uppdatera regressionstesterna

# Logga avvikelser, prioritera och hantera dem

- Logga och beskriv noga
- Tänk på framtida regressioner, beskriv även testmiljön om den inte dokumenterats på annat sätt
- Kan du (får du) gå direkt till den utvecklare som bäst kan åtgärda felet (glöm inte att dokumentera ändå)



# Mäta testprocessen

Testresultat, antal genomförda tester, antal nya buggar och liknande **är alltid ett mått på hur det går.** En lägesrapport.

Men anledningen att ni hittar färre nya buggar ju längre ni testar kan lika gärna visa att testerna är dåliga, som att koden är bra.