

Tentafrågor från grupp C

Uppgift 1, 3p

Enligt IEEE Std har en bra kravspecifikation en mängd fordringar att uppfylla. Kravspecifikationen skall vara;

- A. Korrekt (Correct), det vill säga att varje krav skall reflektera ett behov eller önskemål från kunden.
- B. Komplet (Complete), alla nödvändiga krav skall finnas med.
- C. Entydig (Unambiguous), alla skall vara överens om vad varje krav innebär.
- D. Konsekvent (Consistent), ingen av kravspecifikationens delar skall innehålla någon information som står i konflikt med någon annan del.
- E. Kraven skall vara prioriterade efter hur viktiga och hur stabila de är. (Ranked for importance and stability)
- F. Modifierbar (Modifiable), det vill säga att en kravspecifikation är modifierbar om den är lätt att ändra utan att den förlorar sin konsekvens (consistency).
- G. Verifierbar (Verifiable), ett krav är verifierbart om det finns en möjlighet att kontrollera ifall produkten uppfyller kravet.
- H. Spårbar(Traceable), man skall kunna se var kraven kommer ifrån och hur de används i design och kod.

Dessutom bör en kravspecifikation vara lätt att förstå för kund och utvecklare och man skall kunna spåra de mål kunden har till kraven.

Olika moment i kravhanteringsprocessen och olika sätt att formulera krav, kan hjälpa till i arbetet för att dessa fordringar skall uppfyllas. Ange för följande moment eller kravbeskrivningssätt vilken (1 st) av ovanstående fordringar (A-H) som de i huvudsak hjälper till att uppfylla.

(Rätt svar ger 0,5 p)

Moment/kravformulering	Fordran
1. Uppgiftsbeskrivning som krav (Task description)	A, C, G
2. Händelselista och Funktionslista som krav (Event list and Funktion list)	A, G
3. Tillstånd-övergångs matris som krav(State-Transition matrix)	B, G, (H)
4. Användbarhetstest (Usability test)	B
5. Domän-krav matris (Domain-requirement matrix)	A, B, H
6. Virtual Windows och Virtual Windows test	C, D, G

Rättningsmall:

De svar som är angivna i matrisen kan alla betraktas som rätt av följande anledningar:

Uppgiftsbeskrivning som krav (Task description) är bra att använda då kunden granskar kraven eftersom de skrivs på ett sätt som är lättförståeligt för alla parter (C). Detta gör att det är lättare att kontrollera så att de krav som finns med också reflekterar kundens önsknings och behov (A). Det är även lätt att verifiera att slutprodukten uppfyller dessa krav (G).

Händelselista och Funktionslista som krav (Event list and Funktion list) är bra för att se ifall kraven som är listade uppfyller kundens behov och önsknings (A). Det är också lätt att verifiera att slutprodukten uppfyller listade krav (G).

Tillstånd-övergångs matris som krav (State-Transition matrix) är bra för att hitta de krav som har med just övergångar mellan olika tillstånd att göra. Dessa kan vara ganska svåra att hitta (B). Dessutom är det lätt att verifiera att slutprodukten uppfyller dessa krav (G). Man skulle också kunna tänka sig att det kan vara lättare att se var kraven kommer ifrån och hur de används i design och kod ((H))

Användbarhetstest (Usability test) är nästan ett måste för att man skall kunna fånga upp designkrav och användarkrav (B).

Domän-krav matris (Domain-requirement matrix) är ett bra sätt att spåra hur kraven används i kod och design (H). De hjälper också till för att se till så att alla nödvändiga krav finns med (B) och att visa att kraven beskriver ett behov eller ett önskemål från kunden.

Virtual Windows som krav och Virtual Windows test. Genom att använda sig av Virtual Windows gör man det lätt för alla parter att förstå innebörden av ett krav (C). De gör det också relativt lätt för att verifiera att slutprodukten uppfyller kravet (G). Genom att göra Virtual Windows tester kan man kontrollera så att olika krav inte står i konflikt med varandra (D).

Uppgift 2, 9,5p

Form enligt ”Exempel Påstående - anledning” Föreläsning. 4, s.6. (1/2p per rätt svar)

- A. Både påståendet och anledningen är korrekta uttalanden OCH anledningen förklarar påståendet på ett korrekt sätt.
- B. Både påståendet och anledningen är korrekta uttalanden, men anledningen förklarar inte påståendet.
- C. Påståendet är korrekt, men anledningen är ett felaktigt uttalande.
- D. Påståendet är felaktigt, men anledningen är ett korrekt uttalande.
- E. Både påståendet och anledningen är felaktiga uttalanden.

1. **Påstående:** Många utvecklare anser att deras jobb är att uppfylla de krav som finns. Detta är bara sant om kraven uppfyller kundens förväntningar.
Anledning: Om kraven i kravspecifikationen inte är kontrollerade och validerade är det stor risk för att detta utlöser en konflikt med kunden.
Svar: A, Både Påstående och anledning är sanna. Anledningen förklarar påståendet. En bra kravspecifikation (enligt Enligt IEEE Std) innebär bland annat att kundens krav uppfylls.
2. **Påstående:** Ett av de absolut bästa sätten att se ifall kunden får det system som efterfrågas att kravgranskare gör checklistor (checklists).

- Anledning:** Checklistor är bra att göra för att fånga upp problem och rätta till dessa tidigt i utvecklingsprocessen.
- Svar:** D, Checklistor är inte ett av de bästa sätten att se till så att kunden får det system som efterfrågas. Checklistor används för att kontrollera krav i kravspecifikationen. En korrekt kravspecifikation skulle innebära att alla kundens krav är mötta, men då listor används är det alltid en risk att man missar utelämnade krav. Det skulle vara betydligt bättre att göra tester.
3. **Påstående:** När man vill ha ett anbuds förfarande för att välja leverantör använder man sig ofta av "Pre-qualification".
- Anledning:** Huvud anledning är att man ofta får in bättre anbud om man sollar bort dåliga förslag tidigt.
- Svar:** C, Rätt anledning är att man vill minska andelen anbud till en mängd man klarar av.
4. **Påstående:** Vid ett anbuds förfarande, när man ska bedömma vilket av flera anbud man ska välja bör man ha sedan tidigra bestämda vikter som de olika kriterierna ska multipliceras med.
- Anledning:** Om vikterna inte är bestämda innan så kan olika intresenter påverka valet av dom för eget intresse, t.ex. kan dom få bonus av det företag som vinner anbudet. För att undvika detta så ska man sätta vikterna i förväg.
- Svar:** D, Påståendet är fel eftersom färdig valda vikter ofta inte är så lämpligt, det kan ofta visa sig senare att vissa aspekter inte kom med i vikterna och att ett sämre förslag då ändå kan vinna. Anledning till att man ändå sätter fasta vikter i förväg är den som beskrivs i frågan.
5. **Påstående:** Task & Suport krav är lätta för utvecklare och designers att implementera.
- Anledning:** Eftersom det bara är att följa arbetsuppgiften och implementera efter den.
- Svar:** E, Det är inte alls särskilt lätt att översätta en arbetsuppgift till kod eftersom program ofta inte moduleras efter uppgifter.
6. **Påstående:** Vid marknadsdriven utveckling är det viktigt att hitta beroenden mellan krav.
- Anledning:** Vid planeringen av vilka krav som ska vara med i en realese är det viktigt att kunna säga vilka krav som definitivt kommer att vara med i den kommande realesten, dock så är det inte så viktigt om ett visst krav kommer med i denna eller nästa.
- Svar:** B, Vid marknadsdriven utveckling är det viktigt att veta hur man kan dela in kraven i olika releaser, finns det beroenden så kanske man inte kan dela upp dessa krav i olika releaser. För att kunna marknadsföra en produkt är det viktigt att kunna säga vilka delar som kommer att finnas i releasen men om ett visst krav får vänta en release är inte lika viktigt så länge man leverar det som har lovats tidigare.
7. **Påstående:** Ett installationstest är bra på att testa hur systemet fungerar under lång tid i den verkliga process där systemet kommer att användas.
- Anledning:** Det enda sätt att kunna testa t.ex. hur ofta programet hänger sig eller hur lång tid det tar för en erfaren användare att utföra en viss uppgift är att köra test i den riktiga användingsmiljön

Svar: D, Ett installations test går ut på att bra se ifall programet går att installera och få det att köra, det som beskrivs i text och anledning är ett "Operational test".

8. **Påstående:** För databasdrivna system är skillnaderna ofta få mellan den logiska datamodellen och det som slutligen implementeras.
Anledning: Man tar ofta hänsyn till hur en relationsdatabas fungerar när man gör datamodellen.
Svar: C Påståendet är korrekt, anledningen är felaktig. När man skapar datamodellen tar man inte hänsyn till hur datan kommer att representeras i implementationen, men p.g.a. en relationsdatabas uppbyggnad krävs det inte mycket arbete för att överföra modell i implementation.
9. **Påstående:** Det är svårt att lära sig göra bra datamodeller.
Anledning: Notationen i datamodeller är svår att lära sig.
Svar: C Påståendet är korrekt, anledningen är felaktig. Påståendet stämmer eftersom världen inte är uppenbart indelad i fina lådor och objekt.
10. **Påstående:** Funktions- och händelselistor kan invagga en i falsk säkerhet, eftersom de verkar kompletta och uttömmande.
Anledning: Funktions- och händelselistor är bara ordentligt täckande om man gjort en mycket noggrann elicitering.
Svar: A, Både påståendet och anledningen är korrekta, och anledningen förklarar påståendet på ett korrekt sätt.
11. **Påstående:** Egenskapskrav är svåra för leverantören att implementera.
Anledning: Leverantören måste se till att de uppfyller kundens affärsmål.
Svar: E, Både påståendet och anledningen är felaktiga. Egenskapskrav är lätta att implementera, delvis just för att leverantören *inte* behöver se till att de uppfyller kundens affärsmål.
12. **Påstående:** Leverantörer insisterar ofta på att alla egenskapskrav ska åtföljas av aktiviteter där egenskapen används.
Anledning: Kunder kan få för sig att ställa krav på egenskaper de aldrig kommer att använda.
Svar: A, Både påståendet och anledningen är korrekta, och anledningen förklarar påståendet på ett korrekt sätt.
13. **Påstående:** Beslutstabeller är ofta bra för att beskriva händelselistor.
Anledning: Kunden har (ofta m.h.a. kravspecialisten) goda möjligheter att validera beslutstabeller.
Svar: D, Påståendet är felaktigt, anledningen är korrekt. Beslutstabeller har inte så mycket med händelselistor att göra. Däremot lämpar sig de sistnämnda väl för validering av kund.
14. **Påstående:** Tillståndsövergångsmatriser är ett mycket bra verktyg för att kontrollera att alla situationer och features är påtänkta.
Anledning: Det finns matematiska metoder för att matcha sådana matriser till andra sorters krav.
Svar: C, Påståendet är korrekt, anledningen är felaktig. Några sådana metoder känner vi inte till. Däremot stämmer påståendet, eftersom man vid användandet av

dylika matriser ofta finner obskyra situationer som behöver hanteras.

15. **Påstående:** Det kan vara fördelaktigt att kombinera aktivitetsdiagram och datauttryck.
Anledning: Aktivitetsdiagram är dåliga på att kommunicera datan som överförs mellan olika aktörer.
Svar: A, Både påståendet och anledningen är korrekta, och anledningen förklarar påståendet på ett korrekt sätt.
16. **Påstående:** Quality grid är ett lämpligt sätt att specificera kvalitetskrav.
Anledning: Quality grid gör det enkelt att verifiera att kundens mål är implementerade.
Svar: E. Quality grid är bra för att prioritera vilka kvalitetsfaktorer som är viktiga, men man kan inte specificera krav med hjälp av den eftersom de inte går att verifiera
Inlärningsmål: Frågan relaterar till inlärningsmål 4, 12 och 14 (och lite till 15-17) i föreläsning 1.
17. **Påstående:** Open metric and open target är i praktiken ingen lämplig teknik för att specificera kvalitetskrav.
Anledning: Eftersom leverantören väljer hur kvalitetsfaktorerna ska mätas så leder ofta till att kundens behov inte blir uppfyllda.
Svar: E. Både påstående och anledning är rena lögner. Open target där kunden anger sina förväntningar är en bra ”förhandlingsteknik” för att komma fram till realistiska och ekonomiska gränser för kvalitetskrav. Open metric kan användas för de kvalitetskrav där kunden är osäker på hur de ska mätas.
Inlärningsmål: 6, 12, 15 och 16.
18. **Påstående:** Heuristisk utvärdering anses effektivare än användbarhetstest.
Anledning: Eftersom användbarhetstester kräver ett färdigt system.
Svar: E
19. **Påstående:** Användbarhetstest bör utföras i början av projektet.
Anledning: Eftersom det i projektets slutskede är mycket svårt att rätta felen.
Svar: A

Uppgift 3, 1p

Para ihop rätt krav (1-4) med rätt kvalitetsfaktor (A-K).




- A. Security
- B. Correctness
- C. Reliability
- D. Usability
- E. Performance
- F. Maintainability
- G. Testability
- H. Flexibility
- I. Portability
- J. Interoperability
- K. Reusability

Krav	Kvalitetsfaktor
Om systemet ska kunna köras på en ny processortyp på samma operativsystem så ska max 2% av antalet rader kod behöva ändras	I
Systemet ska vara tillgängligt 99,9% av tiden	C
När fel rättas i produkten så ska antalet relaterade följdfejl vara färre än 0,2	F
Produkten ska klara av 500 transaktioner per sekund under maximal last	E

Uppgiften ger 0,25 poäng per rätt svar, lägre poängutdelning eftersom frågorna är ganska lätta. Uppgiften svarar mot inlärningsmål 3 och 12.

Uppgift 4, 1,5p

Förbind varje krav med respektive namn på stil för användbarhetskrav. Förbind även riskbilderna till rätt krav. Varje riskbild passar bara med ett krav.

Stil	Krav	Risk	
		Customer	Supplier
Förståelsegrad/betyg (Score for understanding)	R1 Produkten ska ge inmatningsförslag (auto complete) vid inmatning av bostadstyp.		
Antal tangentryckningar (Keystroke counts)	R2 Tre prototypversioner ska göras och användbarhetstestas under designen av systemet.	<p>A</p>	
Processkrav (Development process requirements)	R3 Visa 5 användare 10 vanliga felmeddelanden och fråga om orsaken till dem. 80% ska svara rätt.		
Opinionsundersökning (opinion poll)	R4 Max 1 av 5 nybörjare ska stöta på kritiska problem vid utförandet av uppgift A och B.	<p>B</p>	
Uppgiftstid (Task time)	R5 Systemet ska använda skärmbilder beskrivna i appendix xx.		
Designnivå krav	R6 Registreringen av händelse c ska vara möjligt med tre tangentryckningar, ingen mus.	<p>C</p>	
Problemantal (Problem counts)	R7 Menyerna ska ha som mest tre nivåer.		
Riktlinjeföljande krav (Guideline adherence)	R8 85% av användarna ska anse att systemet är lätt att använda.		
Produktnivå krav	R9 Nybörjare ska genomföra uppgift A och B under 15 minuter, erfarna användare under 2 minuter.		

Rättningsmall 0.1p per förbindelse mellan kravstil och krav, 0.2p per förbindelse mellan krav och riskbild.

Problemantal (Problem counts)

R4 Max 1 av 5 nybörjare ska stöta på kritiska problem vid utförandet av uppgift A och B.

Uppgiftstid (Task time)

R9 Nybörjare ska genomföra uppgift A och B under 15 minuter, erfarna användare under 2 minuter.

Antal tangenttryckningar (Keystroke counts)

R6 Registreringen av händelse c ska vara möjligt med tre tangenttryckningar, ingen mus.

Opinionsundersökning (opinion poll)

R8 85% av användarna ska anse att systemet är lätt att använda.

Passar till figur C

Förståelsegrad/betyg (Score for understanding)

R3 Visa 5 användare 10 vanliga felmeddelanden och fråga om orsaken till dem. 80% ska svara rätt.

Designnivå krav

R5 Systemet ska använda skärmbilder beskrivna i appendix xx.

Produktnivå krav

R1 Produkten ska ge inmatningsförslag (auto complete) vid inmatning av bostadstyp.

Passar till figur A

Riktlinjeföljande krav (Guideline adherence)

R7 Menyer ska ha som mest tre nivåer.

Passar till figur B

Processkrav (Development process requirements)

R2 Tre prototypversioner ska göras och användbarhetstestas under designen av systemet.

