

Lunds Tekniska Högskola, Inst. för Datavetenskap

Skriftlig tentamen i ETS170 Kravhantering

Tid: 2010-12-15 kl. 14-19, **Plats:** Eden 22

Hjälpmedel: Inga.

OBS! Tentamen innehåller två delar: Del A Teori 50 poäng, Del B Uppsatsämnen 50 poäng. Del A består av flervalsfrågor och kommer att bedömas schablonmässigt med mallar (ev. automatiskt) och fylls i direkt i detta häfte. Del B innehåller öppna frågor som besvaras i uppsatsform och lämnas in på separata papper.

NAMN: _____

Skriv namn på varje inlämnat papper!

DEL A. TEORI 50p

Denna del innehåller frågor i form av påståenden och anledningar.

För varje par av påstående / anledning, svara med ett av följande alternativ:

- 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.

Alternativen anges med bokstäver A-E i anvisad kvadrat:

Svar A|B|C|D|E|_



Alla kvadrater ska fyllas i med exakt en bokstav. Rätt ifylld ruta ger 1 poäng medan felaktigt ifylld eller ofylld ruta ger 0 poäng.

	<i>Påstående</i>	<i>Anledning</i>	<i>Svar</i> A B C D E _
1	För produktutveckling och inköp av hyllprogramvara (COTS) blir arbetet med kravelicitering snarlikt.	Kunderna och leverantören har i stort sett samma mål.	☐
2	I den yttre domänen finns sällan aktörer som kommunicerar indirekt med systemet.	Aktörerna i den inre domänen kommunicerar med systemet via gränssnitt.	☐
3	Krav på produktnivå är närmare design- än målnivån även då kunden är intern (in-house).	Domän- och målnivåkrav är sällan relevanta i internprojekt (<i>in-house</i>).	☐
4	Beroenden mellan krav är oftast ojämnt fördelade.	Under utgåveplanering (<i>release planning</i>) är beroenden mellan krav viktiga att ta hänsyn till.	☐
5	Ett kontextdiagram är lämpligt om man vill beskriva vilka gränssnitt som finns mellan systemet och dess omgivning.	Kontextdiagram visar en överskådlig bild av direkta aktörer och deras kommunikation med systemet.	☐
6	Uppgiftsbeskrivningar (<i>task descriptions</i>) är ofta lättare för användare att validera jämfört med klassdiagram.	Klassdiagram innehåller sällan entiteter som finns i domänen.	☐
7	I marknadsdriven kravhantering händer det att ett krav går till tillståndet "avslag" (<i>reject</i>) direkt efter initialtillståndet.	Det kan hända att ett krav inte är i linje med den övergripande strategin för mjukvaruprodukten.	☐
8	Spårbarhet (<i>traceability</i>) försvårar ofta underhåll (<i>maintenance</i>).	Kravändringars påverkan på kod kan följas om det finns spårbarhet mellan testfall och design.	☐

	<i>Påstående</i>	<i>Anledning</i>	<i>Svar</i> A B C D E _
9	Vid utveckling med löpande räkning (<i>time-and-material</i>) betalar kunden kostnaderna allt eftersom och priset kan snabbt eskalera.	Det är inte ovanligt att man inte har några skrivna krav vid utveckling med löpande räkning.	☐
10	Följande krav är på produktnivå: "På startskärmen ska det finnas gråa fält för 'användarnamn' och 'lösenord', samt en gul knapp där det står 'logga in'."	Krav på produktnivå kan innefatta in- och utdata.	☐
11	Vid eliciteringen är det ofta bra att fråga intressenter vilka uppgifter de utför.	Användare anger ofta en lösning istället för ett behov.	☐
12	Fokusgrupper är mer strukturerade än idékläckningsmöten (<i>brainstorming</i>).	När man genomför brainstorming uppmanas intressenterna att närsomhelst kritisera idéerna.	☐
13	Tillståndsdigram används med fördel när domänen är mestadels statisk.	Tillståndsövergångar sker som ett resultat av en händelse.	☐
14	Öppen målsättning (<i>open target</i>) låter användaren bestämma det exakta värdet.	Öppet mått (<i>open metric</i>) finns ofta i en dataordlista (<i>data dictionary</i>)	☐
15	CRUD-kontrollen kan med fördel användas för att leta efter saknade eller inkonsekventa krav.	Datakrav är mer lika kvalitetskrav än funktionella krav, varför inkonsekvenser ofta blir mindre riskfyllda.	☐
16	Enligt QUPER uppstår en barriär när kostnaderna ökar drastiskt.	Produkter kan normalt förbättras i små steg till rimlig kostnad, men vid en barriär krävs större investeringar för att nå högre kvalitet.	☐

	Påstående	Anledning	Svar A B C D E _
17	Det är svårt att få ett meningsfullt resultat från ett användbarhetstest om det utförs innan produkten levererats.	Heuristisk evaulering är ofta dyrare än användbarhetstestning, men har mindre risk att identifiera skenbara användbarhetsproblem.	☐
18	För att se om kraven har ett bakomliggande syfte kan spårning till målen tillämpas.	Spårning mellan mål och krav kan både användas som en validerings- och eliciteringsteknik.	☐
19	Releaseplaneringen blir mer trovärdig om man tar hänsyn till kravberoenden.	Releaseplanering som optimeringsproblem blir generellt sett enklare att finna en lösning på med växande antal beroenden och växande antal krav.	☐
20	När man ska bedöma lämplig releasefrekvens bör man ta hänsyn till både kundnyttan och kundernas kostnader för uppdatering.	Kunderna vill i stort sett alltid ha alla nya produkttegenskaper (<i>features</i>), annars brukar de meddela detta spontant.	☐
21	Produktledning för mjukvaruprodukter kräver många skilda kompetenser, t.ex. kravprioritering och releasplanering.	Föreberedelse för lansering (<i>launch preparation</i>) är en kompetens som är extra relevant för de mjukvaruorganisationer som saknar interna intressenter.	☐
22	Varje uppgiftsbeskrivning (<i>task description</i>) bör helst vara måluppfyllande.	Det är bra att sammanföra relaterade deluppgifter i en uppgiftsbeskrivning.	☐
23	Krav på målnivå gör att leverantörer slipper ta ansvar för verksamheten kring produkten.	Vilken kravnivå man väljer beror i stor utsträckning på vem som ska utföra uppdraget	☐
24	Det är ofta lämpligt att ta med både domäninformation och illustrativa exempel i anslutning till ett krav.	Information om bakgrund till kravet samt hypotetiska lösningsförslag ökar chansen att läsaren förstår hur det är tänkt.	☐

	<i>Påstående</i>	<i>Anledning</i>	<i>Svar</i> A B C D E _
25	Om man grupperar kraven i lämpliga grupper kan man minska arbetet med att hantera kravberoenden.	Gruppering av relaterade krav kan ge färre beroenden mellan grupperna än mellan de enskilda kraven.	☐
26	Nuvärdesanalys (net present value analysis) tar inte hänsyn till osäkerheten i framtida vinster.	Framtida vinster är mer osäkra desto längre bort i tiden de inträffar.	☐
27	Öppna mått (<i>open metric</i>) passar bra när man vill överlåta åt leverantören att definiera hur kvalitetskrav ska mätas.	Vissa kvalitetskrav är svåra att kvantifiera och leverantören kanske har specifik domänkunskap inom området.	☐
28	Kvalitetskrav är i praktiken ofta svårare att hantera än datakrav.	Kvalitetskrav rör oftast bara enskilda delar av systemet, medan datakrav oftare påverkar hela systemet.	☐
29	I marknadsdriven kravhantering kan man med fördel använda en databas där kravattributen specificeras successivt.	Kravens förädlingsgrad kan med fördel representeras med en tillståndsmodell.	☐
30	100\$-metoden ger prioriteter som är på en ordinal skala där endast ordningen och inte inbördes storleksrelationer kan avgöras.	Med 100\$-metoden kan man beräkna ett konsekvensindex, då varje jämförelse är med flera gånger.	☐
31	QUPER-modellen gör flera approximationer för att förenkla hanteringen av kvalitetskrav. Bland annat bortser modellen från konkurrenters produktkvalitet.	En barriär når i praktiken inte upp till brytpunkten för differentiering gentemot konkurrenter.	☐
32	Användbarhetsproblem är ofta mer oväntade för utvecklarna än programmeringsproblem.	När man utvärderar ett systems användbarhet försöker man undvika att ta hänsyn till användarnas subjektiva upplevelse.	☐

Påstående	Anledning	Svar A B C D E _
33 En datamodell är bättre på att ange kardinalitet (<i>cardinality</i>) jämfört med virtuella fönster.	En fallgrop med virtuella fönster är att överarbeta den grafiska utformningen, vilket kan öka risken att de tolkas som krav på designnivå.	☐
34 Att ha en speciellt ansvarig för integrationen av en produkt som består av delprodukter från flera leverantörer leder ofta till mer problem än om kunden ansvarar.	Kunden kan ofta själv ta ansvar för integrationen, speciellt om kunden kan verksamheten bättre än programvarutekniken.	☐
35 Att göra parvisa jämförelser mellan krav tar kortare tid än att prioritera varje krav var för sig.	Om kraven prioriteras var för sig får man inte någon information om deras inbördes relation.	☐
36 Kapacitetskrav är ofta enkla att kvantifiera.	Kapacitetskrav kan ofta uttryckas med ett antal eller en storlek.	☐
37 Prestandakrav i fleranvändarsystem är inte lämpliga att uttrycka som en enskild siffra.	Man behöver ofta ta hänsyn till svarstidsfördelningen då medelsvarstiden kan skilja sig stort från svarstiden vid sällsynta överbelastningssituationer.	☐
38 Användbarhetskrav på designnivå utgör en större risk för kunden än för leverantören.	Leverantören kan ganska lätt garantera att designkrav stödjer kundens verksamhet och användarens uppgifter.	☐
39 Varje uppgiftsbeskrivning (<i>task description</i>) bör helst bara ha en aktör.	Det är bättre att dela upp relaterade deluppgifter i olika uppgiftsbeskrivningar, en för varje aktör.	☐
40 Kravhanteringen bör inte sluta förrän kraven är fullständiga.	En fullständig kravspecifikation är lättare att överblicka.	☐

<i>Påstående</i>	<i>Anledning</i>	<i>Svar</i> A B C D E _
41 I en korrekt kravspec återspeglar alla krav existerande behov eller förväntningar.	Behov eller förväntningar räknas normalt inte som krav i begreppets generella betydelse, inkluderande allmänt informationen om vilken mjukvaran som ska utvecklas.	☐
42 Risken för inkonsekventa krav ökar om det finns redundans i kravspecifikationen.	Redundans kan åtgärdas genom att istället göra korsreferenser.	☐
43 Unika nummer för varje krav stödjer modifierbarheten.	Unika nummer underlättar index och korsreferenser, som i sin tur gör det lättare att hitta och hålla reda på krav som ändras.	☐
44 Att hänvisa till standarder hjälper till att undvika en falsk känsla av trygghet.	En hänvisning till en standard bör inte ersätta målnivåkrav som möjliggör verifiering av att standarden uppfyller målen.	☐
45 Krav på utvecklingsprocessen syftar till att ersätta vaga krav med krav på lämpliga arbetsmetoder.	Iterativ utveckling delar upp kravhanteringen i flera steg.	☐
46 Om användbarhetsproblem upptäcks vid systemtest är de ofta lättare att hantera än om de upptäcks vid elicitering.	Användbarhetsproblem kräver ofta smärre förändringar av begränsade delar av användargränssnittet.	☐
47 En fokusgrupp är lämpligare än prototyping för att elicitera realistiska möjligheter.	Arbetet med fokusgrupper liknar brainstorming men är mer strukturerad och sker i flera steg, bland annat ingår framtidsvisioner och prioritering.	☐
48 Intervjuer är sämre än fokusgrupper på att finna krav som rör nuvarande verksamhet snarare än framtida möjligheter.	Mål-domän-analys kan användas både för elicitering och validering.	☐

Påstående

Anledning

Svar

A|B|C|D|E|_

49 Virtuella fönster (*virtual windows*) är ett bra sätt att beskriva ett nytt användargränssnitt.

Det är ofta lättare för användare att hitta saknade datakrav vid validering av virtuella fönster jämfört med E/R-diagram.



50 För hyllprogramvara (COTS) är det mindre lämpligt att ställa krav på designnivå.

Kravhanteringen för hyllprogramvara handlar till stor del om att välja mellan befintliga produkter med redan existerande användargränssnitt.



DEL B UPPSATSER 50p

Utgå från följande rubriker och skriv korta uppsatser på max 2 A4-sidor per uppsats.
Var noga med att skriva läsligt. Svårlästa uppsatser ger poängavdrag.
Börja på nytt blad för varje ny uppsats.

B1. Kravelicitering : barriärer, vad, hur (15 p)

B2. Validering : specifikationskvalitet, vad, hur (15 p)

B3. Prioritering, releasehantering och produktledning vid marknadsdriven kravhantering :
utmaningar, angreppssätt, kompetenser (20 p)

Nr Svar Kap

1 E	1	B1	Elicitering	sum 15	kap
2 D	1	Vad	bra def. Elicitering	max ca 1	1.1
3 C	1	Barriärer	0,5 p per bra beskr av barriär	max ca 4	8.1.1
4 B	[INTDEP]	Vad	0,5 p per bra beskr av "things to elicit"	max ca 3	8.1.2
5 A	3	Hur	1 p per bra beskr av teknik, koppl. vad/barriär	max ca 7	8.2
6 C	4				
7 A	[MDRE2]	B2	Validering	sum 15	kap
8 E	7	Spec.kval.	1 p per bra beskrivning av spec.kval.	max ca 5	9.1
9 A	1	Vad	bra def. Validering	max ca 1	1.1
10 D	1	Vad	exempel kvalitetsegensk. kan vara motsstridiga	max ca 1	9.1
11 B	8	Hur	1-2 p per bra beskriven valideringsteknik:	max ca 8	9.
12 C	8		checklist, CRUD, content, structure, consistency		
13 D	4		checks ag. surroundings, vad/varför		
14 E	6		granskningar, test (prototyp, sim, pilot)		
15 C	6		m.fl.		
16 A	[QUPER]				
17 E	6				
18 B	9	B3	MDRE	sum 20	kap
19 C	[RP1]	utman.	1p per bra utmaning	max ca 10	[MDRE1]
20 C	[RP2]	utman.	utmaning prioritering: skalbarhet, consistency	max ca 2	[PRIO1, PRIO2]
21 C	[SPM]	utm./angr.	rel.pl.: beroenden/constraints, vikta tex stakeh.	max ca 2	[RP1, INTDEP]
22 B	3	angr.	olika priotekniker (AHP, 100\$, betyg...), kontext	max ca 2	[PRIO1, PRIO2]
23 D	1	angr.	laxtrappa, beslutsperspektivet, repository	max ca 2	[MDRE2]
24 A	1	kompetens	koppla ihop delarna till helhet	max ca 2	[SPM]
25 A	[VLSRE]				
26 D	8		p max ca B1-B3 indikerar prioritet och inriktning		
27 A	6		bedömning görs för varje individuellt fall och p är inte strikt		
28 C	1				
29 B	[MDRE2]				
30 E	[PRIO1]				
31 E	[QUPER]				
32 C	6				
33 B	2				
34 E	5				
35 D	[PRIO2]				
36 A	6				
37 A	6				
38 C	6				
39 E	3				
40 E	9				
41 C	9				
42 B	9				
43 A	9				
44 D	3				
45 B	3				
46 E	6				
47 D	8				
48 D	8				
49 D	2				
50 A	1				