

Lunds Tekniska Högskola, Inst. för Datavetenskap

Skriftlig tentamen i ETS170 Kravhantering

Tid: 2008-03-12 kl. 14-19, **Plats:** MA:9A-B

Hjälpmedel: Inga.

OBS! Tentamen innehåller två delar: Del 1 Teori 50 poäng, Del 2 Uppsatsämnen 50 poäng. Del 1 består av flervalfrågor och kryssfrågor. Del 1 kommer att bedömas schablonmässigt med mallar (ev. automatiskt) och fylls i direkt i detta häfte. Del 2 innehåller öppna frågor som besvaras i uppsatsform och lämnas in på separata papper.

NAMN: _____

Skriv namn på varje inlämnat papper!

DEL 1. TEORI 50p

Denna del innehåller frågor som efterfrågar kryss eller bokstäver.



Stämmer bra



Stämmer dåligt

Cirklar i svarsalternativen avser frågor som kräver ställningstagande mellan två alternativ. Ställningstagandet anges med *ett* kryss i *en* av ringarna. Ett korrekt satt kryss ger $\frac{1}{2}$ poäng, ett felaktigt satt kryss ger minus $\frac{1}{2}$ poäng. Om inget av alternativen kryssas ges 0 poäng. Om en fråga innehåller flera ringpar poängsätts dessa var för sig.

Del 1 kan totalt sett inte ge mindre än 0 poäng.

Alternativ



I frågor med kvadrater i svarsalternativet efterfrågas en bokstav. Frågan anger vilka bokstäver som kan användas. T.ex. A-E för olika specificerade alternativ. Alla kvadrater ska fyllas i med exakt en bokstav. Vissa bokstäver kan förekomma flera gånger och det är inte säkert att alla bokstäver behövs. Ibland kan mer än ett alternativ vara rätt. Rätt ifylld ruta ger $\frac{1}{2}$ poäng medan felaktigt ifylld eller ofylld ruta ger 0 poäng.

1A. Påstående/anledning-frågor. (14p)

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.

Påstående: Om en produkt lanseras före konkurrenterna ökar chansen till lönsamhet.

Anledning: Vid tidig lansering har man chans till större marknadsandelar och längre produktlivslängd.

A|B|C|D|E

Påstående: Ett kontextdiagram är inte lämpligt om man vill se vilka gränssnitt som saknas.

Anledning: Kontextdiagram visar vilka direkta aktörer som kommunicerar med systemet.

A|B|C|D|E

Påstående: Vid brainstorming är det viktigt att direkt kritisera orealistiska idéer.

Anledning: Brainstorming är lämpligast vid konsekvens- och riskanalys.

A|B|C|D|E

Påstående: Att göra parvisa jämförelser mellan krav tar längre tid än att sätta prioriteter på varje krav var för sig.

Anledning: Om kraven prioriteras var för sig får man inte så bra information om deras inbördes relation.

A|B|C|D|E

Påstående: En strukturkontroll (structure check) är inte lämplig för att identifiera motstridigheter.

Anledning: Formella språk gör det lättare för lekmän att validera kraven.

A|B|C|D|E



Påstående: Krav som är dyra att implementera tillför ofta ett större värde än de krav som är billiga att implementera.

Anledning: För att uppnå lönsamma produkter är det bättre att maximera nyttan och minimera kostnaden än tvärt om.

A|B|C|D|E



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

Anledning: Det är ofta lättare för användare att hitta saknade datakrav vid validering av virtuella fönster (*virtual windows*) jämfört med E/R-diagram.

A|B|C|D|E



Påstående: I praktiken kan funktionella och icke-funktionella krav vara svåra att särskilja.

Anledning: Uppfyllandet av kvalitetsegenskaper är ofta beroende av speciella funktioner.

A|B|C|D|E



Påstående: Det är oftast säkrast för kunden att huvudleverantören avstår från systemintegration.

Anledning: Den direkta affärsrelationen mellan kunder och underleverantörer sker ofta via internutveckling (*in-house*).

A|B|C|D|E



Påstående: Krav på domännivå innehåller normalt bara klienter från den yttre domänen.

Anledning: Den inre domänen innehåller aktörer som kommunicerar indirekt med systemet via en aktör i den inre domänen.

A|B|C|D|E



Påstående: Kravspecifikationer ska aldrig innehålla designkrav.

Anledning: Designnivåkrav uppkommer ofta då befintliga system ingår i domänen.

A|B|C|D|E



Påstående: Det är inte så vanligt att matematiskt baserade kravtekniker används i industriell systemutveckling.

Anledning: Formella språk gör det svårt för lekmän att validera kraven.

A|B|C|D|E



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

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

A|B|C|D|E



Påstående: Kostnad/värde-relationer estimeras oftast bättre genom relativa bedömningar än med absoluta siffror.

Anledning: Osäkerheterna är ofta stora och både kostnad och värde kan vara svårt att kvantifiera i förväg.

A|B|C|D|E



Påstående: Vid införande av informationssystem i stora organisationer är det lämpligt att genomföra pilotexperiment.

Anledning: Om en stor verksamhet berörs av informationssystemet är införandekostnaden ofta högre än systemutvecklingskostnaden.

A|B|C|D|E

Påstående: Varje uppgiftsbeskrivning (*task description*) bör helst bara ha en aktör.

Anledning: Det är bättre att dela upp relaterade deluppgifter i olika uppgiftsbeskrivningar.

A|B|C|D|E

Påstående: Hot från substitut och nya konkurrenter ingår bland de krafter som kan påverka produktutvecklande företag.

Anledning: Den sena majoriteten (*late majority*) är mindre lönsam jämfört med substitut och konkurrenter.

A|B|C|D|E

Påstående: Det är väldigt viktigt att ta reda på vilka intressenterna (*stakeholders*) för ett projekt är, vilka intressen och attityder de har.

Anledning: Det är från intressenterna som kraven kommer och de behövs för att säkra framgången för ett projekt.

A|B|C|D|E

Påstående: Kravhanteringen bör inte sluta förrän kraven är fullständiga.

Anledning: En fullständig kravspecifikation underlättar kravvalidering.

A|B|C|D|E

Påstående: Det är lämpligt att använda kontextdiagram tidigt i utvecklingen.

Anledning: Den inre domänen innehåller aktörer som kommunicerar direkt med systemet.

A|B|C|D|E



Påstående: Prioritering med betygssättning kan försvåra ändringshanteringen.

Anledning: Vid betygssättning får många krav samma prioritet vilket inte ger en total rangordning och det är därmed inte lätt att avgöra vilka krav som bör strykas först.

A|B|C|D|E



Påstående: Om användbarhetsproblem upptäcks vid systemtest är de ofta svåra att hantera.

Anledning: Användbarhetsproblem kräver ofta små förändringar av begränsade delar av användargränssnittet.

A|B|C|D|E



Påstående: Användbarhet (*usability*) betraktas i standarden ISO9126 som en av många typer av kvalitetskrav.

Anledning: Icke-funktionella krav anger hur bra systemets funktioner är.

A|B|C|D|E



Påstående: Enligt Hofmann och Lehner (2001) var kravprioritering det som ansågs svårast.

Anledning: Det visade sig svårt att involvera kunderna i identifiering av högprioriterade krav.

A|B|C|D|E



Påstående: CRUD-matrisen är bra att använda för att finna krav som saknas i kravspecifikationen.

Anledning: När införandekostnaden är högre än utvecklingskostnaden är det lämpligt att genomföra pilotexperiment.

A|B|C|D|E



Påstående: Varje uppgiftsbeskrivning (*task description*) bör helst vara öppen så att måluppfyllnaden inte är avgjord.

Anledning: Det är bättre att dela upp relaterade deluppgifter.

A|B|C|D|E



Påstående: I uppgiftsbeskrivningar (*task descriptions*) är den exakta ordningsföljden hos deluppgifterna inte nödvändigtvis den enda rätta.

Anledning: För domännivåkrav är uppdelningen mellan vem som gör vad av aktörer och system inte avgjord.

A|B|C|D|E



Påstående: Det är ofta lämpligt att ta med både domäninformation och illustrativa exempel i anslutning till ett krav.

Anledning: Information om syfte och bakgrund till kraven samt hypotetiska lösningsförslag ökar chansen att läsaren förstår hur det är tänkt.

A|B|C|D|E



1B. Vad gäller för dessa påståenden ($\pm 18p$)

Vilken kravnivå man väljer beror i stor utsträckning på vem som ska utföra uppdraget.

- Stämmer bra
 Stämmer dåligt

Genom att fråga "varför" kommer man närmare målnivån. Genom att fråga "hur" kommer man närmare designnivån.

- Stämmer bra
 Stämmer dåligt

Krav på målnivå gör att leverantörer slipper ta ansvar även för omstrukturering av verksamheten kring produkten.

- Stämmer bra
 Stämmer dåligt

Spårbarheten försvåras om alla krav måste numreras.

- Stämmer bra
 Stämmer dåligt

Damian & Chisan (2006) upptäckte att förbättringar i kravhanteringen visserligen ledde till bibehållen kvalitet men inte till förbättrad riskhantering och inte till ökad produktivitet.

- Stämmer bra
 Stämmer dåligt

För att kunna genomföra granskningar krävs nästan alltid ett fungerande system.

- Stämmer bra
 Stämmer dåligt

Vilken kravnivå man väljer beror i huvudsak på typen av användargränssnitt.

- Stämmer bra
 Stämmer dåligt

Vilken kravnivå man väljer beror i huvudsak på intressenternas demografi.

- Stämmer bra
 Stämmer dåligt

I en heuristisk utvärdering låter man en slutanvändare utan tidigare kunskap om systemet utvärdera systemet.

- Stämmer bra
 Stämmer dåligt

Heuristisk utvärdering hittar ofta fler verkliga problem jämfört med användbarhetsutvärdering.

- Stämmer bra
 Stämmer dåligt

Användbarhetsproblem är ofta mer oväntade än programmeringsproblem.

- Stämmer bra
 Stämmer dåligt

En fullständig kravspecifikation kan normalt uppnås med liten ansträngning.

- Stämmer bra
 Stämmer dåligt

En kravingenjör förväntas endast i undantagsfall hjälpa intressenterna att hitta realistiska och kostnadseffektiva produkter.

- Stämmer bra
 Stämmer dåligt

Följande krav:

R1: Leverantören ska tillhandahålla kvalificerad supportpersonal.

är mer riskfyllt för leverantören än för kunden.

- Stämmer bra
 Stämmer dåligt

Följande krav:

R2: Leverantören ska påbörja felrättning inom 24 timmar efter upptäckt.

är mer riskfyllt för leverantören än för kunden.

- Stämmer bra
 Stämmer dåligt

Följande krav:

R3: Leverantören ska rätta alla fel inom 2 veckor efter upptäckt.

är mer riskfyllt för leverantören än för kunden.

- Stämmer bra
 Stämmer dåligt

Följande krav:

R1: Tre prototypversioner ska göras och användbarhetstestas under designen av systemet

är mer riskfyllt för leverantören än för kunden.

- Stämmer bra
 Stämmer dåligt

Följande krav:

R2: 95% av användarna ska anse att systemet är lätt att använda.

är mer riskfyllt för leverantören än för kunden.

- Stämmer bra
 Stämmer dåligt

Följande krav:

R3: Nybörjare ska kunna genomföra uppgift A och B på mindre än 15 minuter, erfarna användare på mindre än 2 minuter.

är mer riskfyllt för leverantören än för kunden.

Stämmer bra

Stämmer dåligt

Ofullständiga datakrav ger mer problem i praktiken än ofullständiga kvalitetskrav.

Stämmer bra

Stämmer dåligt

Enligt Moores modell är kunderna i den tidiga majoriteten (*early majority*) pragmatiska, försöker undvika andras misstag och vill se goda referenser.

Stämmer bra

Stämmer dåligt

När man utvärderar ett systems användbarhet försöker man undvika att ta hänsyn till användarnas subjektiva upplevelse.

Stämmer bra

Stämmer dåligt

Tillståndsdigram kan användas både på domännivå och designnivå, beroende på vad man lägger för betydelse i tillstånden.

Stämmer bra

Stämmer dåligt

Prototypstening (*prototype test*) passar bättre än strukturkontroll (*structure check*) för att hitta realistiska krav.

Stämmer bra

Stämmer dåligt

SLUT-kontroll (skapa, läsa uppdatera, tabort) *CRUD* passar bättre än strukturkontroll (*structure check*) för att hitta inkonsekvenser mellan olika krav.

Stämmer bra

Stämmer dåligt

Pilottest (*pilot test*) passar bättre än strukturkontroll (*structure check*) för att hitta spårbarhets-problem.

Stämmer bra

Stämmer dåligt

Mål-krav-spårning (*goal-requirements tracing*) passar bättre än strukturkontroll (*structure check*) för att hitta saknade krav.

Stämmer bra

Stämmer dåligt

Prototypframställning (*prototyping*) passar bättre än intressentanalys (*stakeholder analysis*) vid elicitering av generella mål och nyckelområden.

- Stämmer bra
 Stämmer dåligt

Prototypframställning (*prototyping*) passar bättre än uppgiftsdemonstration (*task demo*) vid elicitering av problem med det nuvarande systemet.

- Stämmer bra
 Stämmer dåligt

Prototypframställning (*prototyping*) passar bättre än fokusgrupper (*focus groups*) vid elicitering av idéer till framtida system.

- Stämmer bra
 Stämmer dåligt

Prototypframställning (*prototyping*) passar bättre än fokusgrupper (*focus groups*) vid elicitering av realistiska möjligheter.

- Stämmer bra
 Stämmer dåligt

Observation passar bättre än fokusgrupper (*focus groups*) vid elicitering av prioriteter.

- Stämmer bra
 Stämmer dåligt

Idekläckningsmöten (*brainstorming*) passar bättre än observation vid elicitering av nuvarande arbetsuppgifter.

- Stämmer bra
 Stämmer dåligt

Om E/R-diagram är för svåra för intressenterna att tolka kan man med fördel komplettera med virtuella fönster (*virtual windows*).

- Stämmer bra
 Stämmer dåligt

Om E/R-diagram är för svåra för intressenterna att tolka kan man med fördel komplettera med ett klassdiagram.

- Stämmer bra
 Stämmer dåligt

Det är oftast lättare för en slutanvändare att validera händelselistor (*event lists*) på produktnivå jämfört med domännivå.

- Stämmer bra
 Stämmer dåligt

1C. Vad gäller för dessa påståenden jämfört med vad Weidenhaupt m.fl. kommer fram till i artikeln "Scenarios in System Development: Current Practice". (±2p)

Scenarios decrease the complexity of the system

Scenarier minskar systemets komplexitet

Stämmer bra

Stämmer dåligt

Scenarios should preferably not be used in combination with prototypes

Scenarier bör helst inte användas tillsammans med prototyper.

Stämmer bra

Stämmer dåligt

Scenarios prevent agreement and consistency

Scenarier förhindrar samstämmighet och överensstämmelse

Stämmer bra

Stämmer dåligt

Scenarios are static over time

Scenarier utvecklas inte allt eftersom tiden går

Stämmer bra

Stämmer dåligt

1D. Vad gäller för dessa påståenden jämfört med vad Hoffman et al. kommit fram till i artikeln "Requirements Engineering as a Success Factor in Software Projects" (±3p)

Att involvera kunder och användare genom hela kravprocessen är billigare att införa än hantering av spårbarhetsmatriser.

Stämmer bra

Stämmer dåligt

Att prioritera kraven är dyrare att införa än att tillsätta skickliga projektledare och gruppmedlemmar till kravaktiviteter.

Stämmer bra

Stämmer dåligt

Framgångsrika projekt lägger uppåt 30% på kravhantering.

Stämmer bra

Stämmer dåligt

Kravteam som kombinerar prototyper med modellering presterade bättre med avseende på de kvalitetsdimensioner som undersöktes.

Stämmer bra

Stämmer dåligt

Kvaliteten på resultaten från kravhanteringsprocessen (*RE products*) ansågs viktigare än styrning av kravprocessen (*process control*).

- Stämmer bra
 Stämmer dåligt

Framgångsrika kravteam lyckades bättre med att frysa krav tidigt och sämre med att genomföra en uttalad ändringshantering.

- Stämmer bra
 Stämmer dåligt

1E. Enligt Kano-modellen (beskriven av Cohen enligt Karlsson) kan krav delas in i tre olika typer: *normala*, *förväntade* och *sensationella*. Dessa typer avser kravens förmåga att tillfredsställa de olika intressenterna. Ange för varje påstående om detta är rätt [R] eller fel [F] för varje kravtyp. Ett påstående kan vara rätt för flera kravtyper. (6p)

	Normala	Förväntade	Sensationella
Dessa krav är något som en kund eller användare ofta uttryckligen uttalar. De är därför ofta lätta att identifiera.	R F <input type="checkbox"/>	R F <input type="checkbox"/>	R F <input type="checkbox"/>
Dessa krav leder till ökad tillfredsställelse om de uppfylls.	R F <input type="checkbox"/>	R F <input type="checkbox"/>	R F <input type="checkbox"/>
Dessa krav leder till minskad tillfredsställelse om de inte uppfylls.	R F <input type="checkbox"/>	R F <input type="checkbox"/>	R F <input type="checkbox"/>
Dessa krav är oftast outtalade. De är därför ofta svåra att identifiera.	R F <input type="checkbox"/>	R F <input type="checkbox"/>	R F <input type="checkbox"/>

1F. Välj för varje beskrivning den projekttyp som passar bäst. (4p)

Projekttyp

A = Produktutveckling (product development)

B = Internutveckling (in-house development)

C = Kontraktutveckling (contract development)

D = Offertförfrågan (tender)

E = Utveckling på löpande räkning (time and materials)

F = Utveckling utlagd på underleverantör (sub-contracting)

G = Inköp av hyllprogramvara (COTS purchase)

- A|B|C|D|E|F|G Kunden betalar utvecklingskostnaden till leverantören, ofta månadsvis. Kostnaden varierar och slutsumman är ofta inte i förväg känd.
- A|B|C|D|E|F|G Kunden behöver ett system för en viss typisk uppgift och köper det som passar bäst av de system som finns på marknaden.
- A|B|C|D|E|F|G Flera leverantörer får chans att visa vad de kan leverera i ett anbudsförfarande. Denna projekttyp ingår ofta som en lagstadgad del i en offentlig upphandling.
- A|B|C|D|E|F|G Utveckling för en öppen marknad där marknadsavdelningen har kundkontakt och samtidigt ofta agerar intern kund åt utvecklingsavdelningen.
- A|B|C|D|E|F|G En avgränsad del av utvecklingen lämnas vidare till en tredje organisation. En integratör ansvarar sedan för helheten och leveransen till kunden.
- A|B|C|D|E|F|G En uppdragsgivare och en leverantör reglerar genom styrdokument vad som ska levereras, ofta innefattande kundspecifik utveckling med överenskommen prismodell.
- A|B|C|D|E|F|G Utvecklingen sker för eget bruk och baseras inte på kontrakt mellan skilda juridiska parter. Ofta genomförs utvecklingen utan kravspecifikation, inte sällan med ödesdigra följder.
- A|B|C|D|E|F|G Denna typ av utveckling kännetecknas av många kunder och konkurrenter, evolution genom releaser, och fokus på lönsamhet och marknadsandelar.

1G. Damian & Chisan (2006) undersökte följderna av förbättringar i kravhantering. Para ihop följderna A-G med rätt förbättringsområden. (3p)

A = minskat omarbete (*reduced rework*)

B = effektivare kommunikation (*more effective communication*)

C = noggrannare estimat (*more accurate estimates*)

D = bättre täckning av produktgenskaper (*improved feature coverage*)

E = bättre hantering av smygande kravökningar (*managed requirements creep*)

F = effektiv förhandling om projektomfång (*effective project scope negotiation*)

G = färre fel (*fewer defects*)

Nedbrytning av produktgenskaper, storleksbedömning och ändringshantering ledde till
Feature decomposition, sizing, and change management led to

A|B|C|D|E|F|G

Från början definierade testscenarier, kravvalidering och granskningar ledde till
Upfront test scenario definition, requirements validation, and peer-reviews led to

A|B|C|D|E|F|G

Ökad förståelse för produktgenskaper, ändringshantering och projektuppföljning ledde till
Enhanced feature understanding, change management, and project tracking led to

A|B|C|D|E|F|G

Ändringshantering och storleksbedömning av produktgenskaper ledde till
Change management and feature sizing led to

A|B|C|D|E|F|G

Spårbarhetslänkar, granskningar och kravvalidering ledde till
Traceability links, peer-reviews, and requirements validation led to

A|B|C|D|E|F|G

Gemensamma grund och tvärfunktionella grupper i utvecklingen ledde till
Common ground and cross-functional teams in feature development led to

A|B|C|D|E|F|G

DEL 2 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.

2A. Typiska svårigheter vid kravelicitering (10 p)

2B. Den goda kravspecifikationen: ideala egenskaper och praktiska hänsyn (15 p)

2C. Icke-funktionella krav: utmaningar, kategorier, exempel och tekniker (15 p)

2D. En jämförande diskussion om marknadsdriven och kontraktbaserad kravhantering (10 p)

FACIT del 1:

1a:ADEBCDDAEEDAAAAECAEBACBABEAA
1b:BBDDDDDDDBDDDDDBDBBDBBBDBDDDBDDDBDD
1c:BDDD
1d:BDBBBD
1e:RFFRFRRRFFRR
1f:EGDAFCBA
1g:CDEFGB

Bedömning del 2:

2A. Lauesen sid 331-335: (10p)

1p per bra förklaring av nedan punkter

1. Svårigheter att uttrycka behov
2. Svårt att förklara vad man gör (idag) och varför
3. Frågar efter specifika lösningar
4. Svårt att föreställa sig nya lösningar
5. Svårt att föreställa sig konsekvenser (av nya lösningar)
6. Motstridiga krav
7. Motstånd till förändringar
8. Onödiga "lyxkrav"
9. Nya krav kommer först efter att andra krav uppfylls
10. Allmänt om elicitering: stegvis, iterativt, när sluta?

2B. Lausen sid 373-380, Karlsson1 sid 67-77 (15p)

1 p (max 10p) per bra förklaring av nedan. (En halv poäng om förklaring saknas, noll om fel):

1. Korrekt
2. Fullständig
3. Otvetydig
4. Konsekvent
5. Rankad/prioriterad för viktighet/stabilitet etc.
6. Modifierbar (lätt att ändra)
7. Spårbar tillbaka till mål, framåt till design/kod
8. Spårbar från mål och framåt till krav.
9. Begriplig
10. Icke-redundant
11. Verifierbar

Mellan 0 och 5 p beroende på hur bra diskussion om avvägningar mellan några av ovanstående kvaliteter samt visad insikt om hur begränsade resurser innebär svårigheter att uppnå perfektion. Relatera till problemet: När vet man att man är färdig med kravhantering?

2C. Lauesen sid 217-284 (15p)

0-2p för definition av kvalitetskrav(icke-funktionella krav), etc.

0-4p för utmaningar: exempelvis * beskrivning att NFR ofta berör hela systemet * avvägningar * glidande skala * svårt att separera från funktioner i praktiken, etc.

0-6p för bra beskrivning av kategorier sid 220-224

0-3p för exempel på tekniker sid 226-284

2D. Karlsson2 sid 5-7 (10p)

1p per bra förklaring av raderna i tabell 1: * antal kunder * antal konkurrenter * valfrihet innehåll * valfrihet leveranstid * valfrihet prissättning * huvudmål (TTM vs uppfylla spec.) * Framgångskriterier:

(försäljning/marknadsandelar/lönsamhet vs acceptans/tillfredsställelse) * releasemönster * graden av formalitet * kända/okända kunder * kundnärhet * process (strid ström/prioritering/kostnadsestimering/releaseplanering vs elicitering/modellering/validering/konflikthantering) * långsiktighet.