

Niklas Hallberg, Per-Ola Lindell, Sofie Pilemalm, Maria Andersson, Leni Ericson

Kravhantering för FMA

TOTALFÖRSVARETS FORSKNING SINSTITUT

Ledningssystem
Box 1165
581 11 Linköping

FOI-R--1503--SE

December 2004

ISSN 1650-1942

Underlagsrapport

Niklas Hallberg, Per-Ola Lindell, Sofie Pilemalm, Maria Andersson, Leni Ericson

Kravhantering för FMA

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystem Box 1165 581 11 Linköping	Rapportnummer, ISRN FOI-R--1503--SE	Klassificering Underlagsrapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år December 2004	Projektnummer E781042
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 41 Ledning med samband och telekom och IT-system	
Författare/redaktör Niklas Hallberg Per-Ola Lindell Sofie Pilemalm Maria Andersson Leni Ericson	Projektledare Niklas Hallberg	
	Godkänd av	
	Uppdragsgivare/kundbeteckning	
	Tekniskt och/eller vetenskapligt ansvarig Niklas Hallberg	
Rapportens titel Kravhantering för FMA		
Sammanfattning (högst 200 ord) <p>Försvarsmakten satsar stora resurser på utveckling av informationssystem, såväl av nya som uppdateringar av befintliga system. Svårigheter med att utveckla <i>rätt system</i>, till <i>rätt pris</i> och i <i>rätt tid</i> är sedan länge ett välkänt fenomen, orsakat bland annat av bristfällig kravhantering. Utvecklingen av en arkitektur ställer större krav på en adekvat kravhantering än vad andra typer av informationssystem gör. Anledningen är att en arkitektur är svårare och mer resurskrävande att modifiera då den väl har blivit införd.</p> <p>Syftet med denna rapport är att ge underlag för utveckling av ett stöd för kravhantering för FMA. Underlaget består av fyra delar: en kravhanteringsprocess, en modell för att beskriva krav, ett datorstöd för kravhantering, och en process för att identifiera nya och validera befintliga arkitekturkrav.</p> <p>Kravhanteringsprocessen består av de tre delprocesserna verksamhetsanalys, behovsanalys och kravanalys. Modellen beskriver krav, kravattribut samt relationer till källor och behov. Datorstödet skall hantera intressenter, källor, behov, krav, samt stödja prioritering av behov och krav, visualisera krav och generera rapporter. Process för att identifiera nya och validera befintliga arkitekturkrav baseras på scenarion och prototyper, som utformats i det verksamhetsnära utvecklingsarbetet.</p>		
Nyckelord Kravhanteringsprocess, FMA		
Övriga bibliografiska uppgifter	Språk Svenska	
ISSN 1650-1942	Antal sidor: 23 s.	
Distribution enligt missiv	Pris: Enligt prislista	

Issuing organization FOI – Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 Linköping	Report number, ISRN FOI-R--1503--SE	Report type Base data report
	Programme Areas 4. C4ISR	
	Month year December 2004	Project no. E781042
	General Research Areas 5. Commissioned Research	
	Subcategories 41 C4I	
Author/s (editor/s) Niklas Hallberg Per-Ola Lindell Sofie Pilemalm Maria Andersson Leni Ericson	Project manager Niklas Hallberg	
	Approved by	
	Sponsoring agency Defence Materiel Administration	
	Scientifically and technically responsible Niklas Hallberg	
Report title (In translation) Requirements engineering for FMA		
Abstract (not more than 200 words) <p>The Swedish armed forces invest large resources on the development of information systems, new as well as modifications of existing systems. The difficulties with development of the adequate systems, to the adequate costs, and in the right time are well-known phenomena, caused by, among other things, insufficient requirement engineering. The development of architectures puts larger demand on adequate management of requirements since it will be more difficult to modify after introduction than other kinds of information systems.</p> <p>This report constitutes a basis for the development of a support for management of requirements for FMA. The support consists of four parts, which are a requirements engineering process, a model to describe requirements, a computer based support for management of requirements, and a process for identification of new requirements and validation of existing requirements. The requirement engineering process includes the three sub-processes business analysis, needs analysis, and requirements analysis. The model describes requirement attributes, and relations to sources and needs. The computer support should handle stakeholders, sources, needs and requirements. It should also handle prioritizations of needs and requirements as well as visualization of requirements and generation of reports. The process for identification of new and validation of existing architecture requirements is based on scenarios and prototypes, which have been designed in close relation to the business oriented development.</p>		
Keywords Requirements engineering process, FMA		
Further bibliographic information	Language Swedish	
ISSN 1650-1942	Pages 23 p.	
	Price acc. to pricelist	

INNEHÅLL

1	Inledning	5
2	Syfte.....	5
3	Kravhantering.....	6
3.1	Kravhantering som en process.....	7
3.2	Ansatser till kravhantering.....	9
4	Genomförande.....	10
4.1	Process för kravhantering	10
4.1.1	Teoretisk utgångspunkt.....	11
4.2	Modell för beskrivning av krav.....	13
4.2.1	Teoretisk utgångspunkt.....	13
4.2.2	Datorstöd för kravhantering.....	15
4.2.3	Teoretisk utgångspunkt.....	16
4.3	Process för analys av krav.....	19
4.4	Teoretisk utgångspunkt.....	19
5	Benämningar, akronymer och förkortningar	21
6	Referenser	22

1 Inledning

Inom Försvarsmakten läggs stora resurser i form av tid och kostnader ner på utveckling av informationssystem. Utvecklingen av nya system såväl som uppdateringar av befintliga system kommer framöver att vara en kontinuerlig verksamhet. Förväntningarna på nyttan av dessa system är högt ställda. För att förhoppningarna skall infrias till rimliga kostnader och inom rimlig tid, krävs modeller och metoder för att säkerställa att användarnas och verksamhetens behov avspeglas i utvecklade system. Svårigheter med att utveckla *rätt system*, till *rätt pris* och i *rätt tid* är dock sedan länge ett välkänt fenomen. Redan under mitten av 60-talet påtalades svårigheter och brister inom systemutveckling. Många informationssystem motsvarar inte användares och verksamhetens behov (Brooks, 1995). De system som nått upp till en acceptabel nivå mot ställda förväntningar har oftast blivit väsentligt dyrare än beräknat och levererats betydligt senare än utlovat. Dessa svårigheter har kvarstått trots intensiva försök att komma tillrätta med dem (Young, 2001).

Utvecklingen av informationssystem inom Försvarsmakten är inte förskonade från svårigheterna inom systemutveckling. Aktuell forskning har tvärtom visat på att problem relaterade till Försvarsmaktens systemutvecklingsprojekt är än flera och större (Morin, 2002; Persson, 2000). Dessa problem innefattar exempelvis svårigheter med att utveckla system som passar alla de situationer som kan uppstå och som måste kunna hanteras av informationssystemet samt svårigheter att förena dynamik och krav på makt, autonomitet och kontroll med krav på stabilitet (Morin, 2002).

En grundläggande orsak till svårigheter i systemutveckling generellt är avsaknaden av stöd för hantering av krav. Kravhantering innefattar identifiering av verksamhetens och användares behov och överföring av dessa behov till konkreta och dokumenterade krav. Inom systemutvecklingsområdet är det först på senare år som vikten av en adekvat kravhanteringsprocess, som i realiteten utgår från verksamhetens och användares behov, har uppmärksamats (Siddiqi & Shekaran, 1996). En förståelse för verksamheten och behovet av stöd så tidigt som möjligt i systemutvecklingsprocessen är central för att slippa kostsamma omarbetningar. Att system utvecklade för Försvarsmaktens räkning har en adekvat kravhanteringsprocess där det färdiga systemet verkligen bygger på en riktig kravbild är dessutom fundamentalt, eftersom konsekvenserna annars kan blir förödande.

2 Syfte

Syftet med denna rapport är att ge problembeskrivning av kravhantering samt utgöra ett underlag för hur ett stöd för kravhantering för Försvarsmaktens Arkitektur (FMA) skall utvecklas. Kravhanteringsprocessen utvecklas således specifikt för FMA men är även avsedd att vara generellt applicerbar.

Underlaget för hur ett stöd för kravhantering skall utvecklas delas upp i fyra delsyften gällande:

1. Kravhanteringsprocess.
2. Modell för att beskriva krav, prioritering av krav samt kravens relationer till kravställare och kravställares behov.
3. Ett datorstöd för kravhantering som baseras på kravhanteringsprocessen och modellen för att beskriva krav.
4. En process för hur prototyper kan användas för att identifiera och validera arkitekturkrav.

3 Kravhantering

Kravhantering refererar till den del i systemutvecklingsprocessen där användarbehov och korresponderande krav på system identifieras, analyseras, specificeras och dokumenteras, innan faser av design, kodning och implementation tar vid (Dorfman, 1996). Begreppet krav (requirements), definieras av Kotonya och Sommerville (1998) som ett uttalande om en systemservice eller en begränsning av systemet. Young (2001) menar att krav är liktydigt med ett nödvändigt attribut i ett system eller ett uttalande som identifierar en förmåga, karakteristik eller kvalitativa faktorer i ett system, för att det ska kunna ha värde och vara till nytta för användaren. Krav kan också ses som specifikationer av vad som ska implementeras i systemet (Sommerville & Sawyer, 1996). De är beskrivningar av hur systemet ska uppträda, eller av en systemegenskap eller ett attribut.

Kravhantering är ett vitt kunskapsområde inom vilket många processer, metoder och tekniker utvecklats, exempelvis objektorienterade ansatser som Rational Unified Process (RUP), prototypansatser som Rapid Application Development (RAD), och socio-tekniska ansatser som ETHICS (Jacobson, Booch & Rumbaugh, 1998; Kotonya & Sommerville, 1998; Graham, 1998; Dorfman, 1996; Mumford, 1995). Centrala aspekter i kravhanteringsprocessen är dock att: (1) identifiera verksamhetens och användares behov av stöd, (2) prioritera dessa behov utifrån verksamhetens och användarnas perspektiv, (3) överföra identifierade behov till krav på informationssystemet, samt att prioriteringar av behov överförs till prioriteringar på krav, (4) spårbarhet mellan krav och verksamheten, (5) förmedla dessa krav på systemet till dem som skall designa systemet, samt (6) skapa underlag för validering och verifiering.

Brister inom kravhanteringsprocessen är den vanligaste orsaken till misslyckad systemutveckling (e.g. Young, 2001). Eftersom varje systemutvecklingsprojekt med avseende på utgångsläge, målsättning och involverade intressenter är unikt behövs en analys av vilka förutsättningar och tillvägagångssätt som är bäst lämpade i varje enskilt fall.

Ett tidigt synsätt på kravhantering präglades av ett produktfokus där systemets grad av uppfyllande av krav uttryckt i specifikationer sågs som ett mått på framgång. Kravhantering sågs som en avgränsad aktivitet i systemutvecklingsprocessen vars enda koppling till övriga aktiviteter förmedlades via en kravspecifikation. Möjligheten att iterera kraven uteblev därmed till stora delar samt att förutsättningar saknades för att validera kraven och verifiera utvecklade system mot krav. Den traditionella synen på kravhantering resulterade i färdiga system av otillräcklig kvalitet. Bristfällig återkoppling från användare och verksamhet under kravhanteringen ledde till att kraven på systemet inte svarade mot verksamhetens och användarnas reella behov (Kotonya & Sommerville, 1998). Produktfokus ledde till att kopplingen mellan behov och krav var bristfällig och systemens nytta och användbarhet var långt ifrån optimal. Det ensidiga produktfokus ledde dessutom ofta till otillräcklig dokumentation eftersom det tenderade att blicka framåt på produkten och negligera själva systemutvecklingsprocessen. Detta medförde i sin tur problem med spårbarhet mellan behov, krav och design samt till systemkrav som var inkonsistenta eller ofullkomliga (Kotonya & Sommerville 1998; Buckingham Shum, 1996). Utebliven iteration, validering, verifiering och möjlighet till modifiering av krav resulterade vidare till statiska krav, vilket innebar att det vid systemleverans inte alltid var de relevanta kraven som realiserats i systemet.

3.1 Kravhantering som en process

Kravhantering har under senare år gått mot ett mer processororienterat fokus som även beaktar organisatoriska och sociala aspekter. Mer utrymme har givits till iteration där krav identifieras, valideras och modifieras genom hela systemutvecklingsprocessen (Robertson & Robertson, 1999). Såväl hela kravhanteringsprocessen som dess ingående delprocesser kan itereras (Young, 2001). En kravhanteringsprocess kan i generella termer beskrivas som innefattande följande steg (Sommerville & Sawyer, 1996):

- *Kravidentifikation.* Detta steg innefattar exempelvis identifiering och definiering av användargrupper och övriga grupper av intressenter samt kontakt med och datainsamling från dessa. Ur det insamlade materialet identifieras verksamhetens och användarnas behov.
- *Kravanalys.* De identifierade behoven analyseras och överförs till krav. Steget innefattar även en prioritering av kraven samt förhandlingar och kompromisser mellan olika intressenter för att enas om gemensamma krav.
- *Kravvalidering.* Kraven valideras, kontrolleras med avseende på konsekvens och fullständighet, modifieras och verifieras. Detta steg kan innehålla stora delar iteration där exempelvis prototyper används för att validerar krav, men iteration kan ske över hela processen.

En processororienterad ansats till kravhantering innebär också att kravhanteringsprocessen i sig inte ses som en enskild aktivitet i den övergripande systemutvecklingsprocessen utan som integrerad med övriga delar i denna. Det innebär att den kravspecifikation som lämnas över då faser av design, kodning och implementation tar vid, inte bara bör vara lättillgänglig och kommunicerbar gällande de ställda kraven på produkten. Den bör även reflektera hela kravhanteringen genom att tydligt påvisa kopplingar till intressenter, behovsunderlag och behov. Spårbarhet måste finnas mellan de olika faserna så att eventuella förändringar kan ske dynamiskt under hela systemutvecklingsprocessen.

Trots den processororienterade ansatsen för kravhantering har flera problem relaterade till den traditionella synen kvarstått, om än i lägre grad. Kravhanteringen kritiseras ofta för att vara otillräcklig då det gäller att fånga upp användares reella situation, problem och behov och för brist på aktiv användarmedverkan (Schuler & Namioka, 1993). Ofta har svårigheterna att samla in och tolka behov av stöd för att genomföra verksamhet negligerats. Fortfarande saknas beskrivningar av hur information om användares önskemål bör inhämtas eller hur systemutvecklare skall "lära känna användaren" (know the user) (Carlshamre, 2001). Vanligt är också brister i kommunikation mellan systemutvecklare och användare (Sommerville & Sawyer, 1996).

Systemutvecklare utan kunskap om domänen där systemet skall användas kan inte rakt av designa och implementera ett system enbart på data insamlad om användares önskemål och beskrivningar. Dessa måste först formuleras till behov för att sedan ytterligare omformuleras och transformeras till krav. Dessa krav måste formuleras på ett sätt så att de som skall ta fram lösningsförslag och implementera dessa har ett tydligt underlag, inga möjligheter till misstolkningar får finnas. Fortfarande saknas dock till stor del konkreta metoder och verktyg som stöder denna process att omvandla önskemål, först till behov, sedan till krav på systemet.

Verksamhetsperspektivet utelämnas ofta trots att verksamheten som sådan inte bara har önskemål och behov, utan även olika typer av restriktioner på systemet. Trots att krav inte längre betraktas som statiska har många ansatser behållit en statisk syn på verksamheten, både före och efter införandet av systemet. Organisatoriska konsekvenser som ett resultat av systemets införande har alltså inte beaktats som en del i kravhanteringsprocessen (Ditsa & Davies, 2000; Kuutti, 1999). Båda fallen kan resultera till system som inte är kompatibla med rådande organisatorisk kultur, system som förblir oanvända eller system som får oönskade organisatoriska konsekvenser.

Andra kvarstående problem då det gäller kravhantering innefattar exempelvis hur spårbarhet erhålls mellan krav och verksamhet. Spårbarhet mellan krav och verksamheten är en förutsättning för en förståelse av vilka krav som systemet

måste uppfylla. Spårbarheten ger också möjligheten att validera identifierade krav mot behov i verksamheten.

Många kravhanteringsansatser är även bristfälliga gällande prioriteringen av behov samt hur dessa prioriteringar sedan överförs till prioriteringar av kraven. Alla behov av stöd som finns i en verksamhet kan inte implementeras i ett informationssystem och framför allt inte med samma ambitionsnivå. Behoven bör istället prioriteras med avseende på vad i verksamheten som är viktigast att stödja.

Kravdokumentation utgör ett underlag för validering av att rätt krav identifierats samt verifiering av att rätt funktionalitet implementerats. När det gäller valideringen så bidrar spårbarhet mellan krav, behov och behovsunderlag (önskemål) till möjligheten att kontrollera kravs berättiganden. Kravhanteringsprocessen bör således vara dynamisk där exempelvis nya identifierade behov direkt kan spåras till förändringar i kravbilden. När det gäller verifiering utgör kraven grunden för vad systemet ska verifieras mot. Trots vikten av en adekvat dokumentation av kraven så är den än idag oftast otillräcklig. Detta resulterar i sin tur i otillräcklig dynamik i kravhanteringsprocessen där spårbarhet, exempelvis mellan behov och krav saknas, vilket kan leda till att det inte är den korrekta kravbilden som presenteras i kravs-specifikationen.

Återkoppling från användare ges vanligen otillräcklig uppmärksamhet. Det är önskvärt att så tidigt som möjligt i systemutvecklingsprocessen få återkoppling från användare för att slippa omarbetningar senare i processen. Deras medverkan i kravhantering ger validering och kvalitetssäkring av slutprodukten. Tidig återkoppling från användare och övriga intressenter medför att utvecklingstiden kortas och därmed minskas kostnaden för utvecklingen.

3.2 Ansatser till kravhantering

Det finns idag ett stort antal ansatser för att möta ovanstående centrala aspekter av att komma till rätta med brister i kravhantering. Olika ansatser har sina styrkor inom olika delar av kravhanteringen. Ansatser från sociologin, exempelvis etnologi, har länge påtalat vikten av att tidigt i utvecklingsarbetet ta hänsyn till användarna som agenter i den verksamhet de befinner sig i. Deltagande designansatser har påvisat hur användare skall involveras att aktivt delta i systemutvecklingsprocessen, och inte endast agera som informationskällor (Schuler & Namioka, 1993). Ansatser med ursprung ur psykologin har visat hur hänsyn till människors psykiska kapacitet, styrkor och svagheter, skall tas. Detta har framförallt gjorts i utvecklingen av interaktionen mellan människor och tekniska system.

En teknisk ansats som på senare tid väckt stort intresse är Unified Model Language (UML). UML utvecklades ursprungligen för att stödja

Objektorienterad systemutveckling men har även rönt stort intresse när det gäller modellering av verksamhet, där notationen Use Cases (svenska användningsfall) har utnyttjas som en konceptuell brygga mellan användarrepresentanter och systemutvecklare. Ansatser från kvalitetsteknikområdet, exempelvis Total Quality Management (TQM) och Quality Function Deployment (QFD) (Braithwaite, 1993), har visat på möjligheten att skapa koherenta systemutvecklingsprocesser och hur intressenters behov skall fås att styra utvecklingsarbetet. Mer verksamhetsorienterade ansatser, exempelvis Business Process Reengineering (BPR), har visat på vikten av att systemutveckling omfattar både teknikutveckling och verksamhetsutveckling (Davenport, 1993). För att maximera nytta av nya informationssystem måste deras utveckling integreras med en verksamhetsutveckling. Flertalet av ovanstående ansatser har och kommer beaktats i genomförandet av detta projekt, för att reducera de problemen som finns relaterade till kravhantering.

4 Genomförande

Utveckling av stöd för kravhantering beskrivs i relation till varje delsyfte (kravhanteringsprocess, modell för krav, datorstöd för kravhanteringen och process för att identifiera och validera krav) tillsammans med ett teoretiskt underlag/utgångspunkt för genomförandet.

4.1 Process för kravhantering

Syftet är att skapa en process för kravhantering. Processen utvecklas främst för att identifiera kraven på FMA, men ska vara generellt applicerbar så att den kan användas för kravhantering av system baserade på FMA. Utvecklingen av processen sker i tre steg, *Definition av process*, *Identifiering av metoder och tekniker* och *Utvärdering av processen*. Dessa steg itereras kontinuerligt för att förbättra processen.

Definition av process

ISO/IEC 15 288 är en standard för systemutveckling som är under bearbetning. I ISO/IEC 15 288 definieras en process för kravhantering bestående av de tre delprocesserna *verksamhetsanalys*, *behovsanalys* och *kravanalys* (ISO/IEC 15 288, version 4, 2001). Var och en av dessa delprocesser innefattar ett antal aktiviteter. ISO/IEC standarden används som utgångspunkt när denna process definieras.

Identifiering av metoder och tekniker

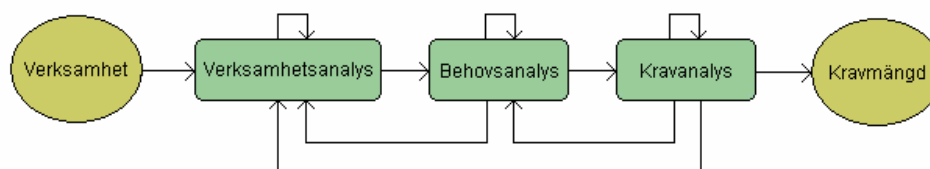
För att stödja genomförandet av definierade aktiviteter i processerna behövs metoder och tekniker. Dessa identifieras i vedertagen litteratur från ett flertal kunskapsdiscipliner.

Utvärdering av processen

För att utvärdera processen appliceras den i ett antal fallstudier, där data om utfallet samlas in med stöd av intervjuer och deltagande observationer. Resultat av utvärderingen används för att förbättra kravhanteringsprocessen.

4.1.1 Teoretisk utgångspunkt

Processen som används som underlag för utvecklingen av kravhanteringsprocessen är iterativ, Figur 1. Det vill säga att i varje steg analyseras om tillräcklig mängd indata finns och om inte så efterfrågas detta från föregående steg. För varje delprocess finns ett antal aktiviteter identifierade, dessa skall stödjas med vedertagna metoder och tekniker.



Figur 1 Kravhanteringsprocessen

Verksamhetsanalys

Målet med verksamhetsanalysen är ett *Behovsunderlag*. Behovsunderlag är det första utvecklingsstadiet för krav och utgörs av information insamlad om den aktuella verksamheten och intressenternas direkt och indirekt formulerade krav på systemet. Behovsunderlag dokumenteras skriftligen på det sätt de inhämtats utan att några omformuleringar görs. Dokumenten benämns kravställande dokument. Första steget i framtagandet av behovsunderlaget är att skaffa sig kunskap om aktuell verksamhet och problemdomän. Utifrån den erhållna kunskapen identifieras systemets intressenter, d v s de som på något sätt kommer att beröras av systemet. Samtliga intressenter kan inte alltid ges tillfälle att påverka systemet. I så fall görs en prioritering av intressenter baserat på hur centrala de anses vara för systemutvecklingen. I nästa steg insamlas och tolkas intressenternas önskemål om systemet och de restriktioner som finns på systemet. En rad datainsamlingsmetoder kan användas för detta, både traditionella datainsamlingsmetoder (exempelvis intervjuer, enkäter, litteraturstudier, deltagande observation) och systemutvecklingsmetoder som bygger på aktiv användarmedverkan (exempelvis dramatisering, Future Workshops). Dessutom beaktas aspekter av Business Process Reengineering och förändring för att komplettera systemutveckling och införande av system med nödvändiga eller önskvärda förändringar i verksamheten. Mer specifikt är aktiviteterna i verksamhetsanalysen att:

- Beskriva verksamheten
- Beskriva problemdomänen
- Beskriva organisatoriska/politiska intressen

- Identifiera intressenterna
- Eventuellt prioritera intressenter och definiera de som direkt bör påverka systemet
- Samla in intressenternas funktionella önskemål
- Samla in intressenternas restriktioner
- Definiera systemets driftsmiljö
- Definiera domänrestriktioner

Vid insamlingen registreras källan (intressent, företagsstandard, doktriner mm) för önskemål och restriktioner för att säkra spårbarheten. Varje önskemål och restriktion ges också en logisk grund för dess existens.

Behovsanalys

Målet med behovsanalysen är att intressenternas verkliga *behov* identifieras ur insamlat behovsunderlag samt att behoven analyseras och prioriteras för att skapa en korrekt behovsbild. Behov utgör det andra utvecklingsstadiet för ett krav. Behoven dokumenteras som användningsfall. Mer specifikt är aktiviteterna i behovsanalysen att:

- Identifiera behov ur insamlat behovsunderlag
- Analysera behoven
- Fastställa systemgränser
- Validera behov med intressenter
- Prioritera behov, med utgångspunkt från ett intressentperspektiv

Behoven formuleras som användningsfall och bildar en första användningsfallsmodell av systemet. Insamlade restriktioner relateras till de användningsfall de begränsar. Samtliga användningsfall och restriktioner tilldelas referenser till sina källor för att säkra spårbarheten. För prioritering av behov används metoder och verktyg hämtade från Japanskt kvalitetstänkande såsom Total Quality Management (TQM) och Quality Function Deployment (QFD).

Kravanalys

Målet med kravanalysen är att formulera *krav* med utgångspunkt utifrån behoven och insamlade restriktioner. Krav är det tredje och avslutande utvecklingsstadiet för ett krav. Krav skall dokumenteras på så sätt att de är validerbara, verifierbara, förståeliga och entydiga. Mer specifikt är aktiviteterna i kravanalysen att:

- Modellera kraven
- Formulera kravtexter
- Analysera kraven med avseende på:
 - *Konflikter*
 - *Överlappningar*
 - *Utelämnningar*

- *Inkonsistenser*
- Lösa problem och konflikter med intressenter
- Transformera prioriteringar
- Fastställa risker
- Säkra spårbarhet mellan önskemål, behov och krav
- Bestämma verifieringsmetod

Med utgångspunkt från användningsfallsmodellen skapas en kravmodell baserad på aktivitetsdiagrammen i UML. Varje funktionellt krav refereras av ett användningsfall och varje icke-funktionellt krav refereras av ett användningsfall, systemenhet eller aktör. Att åstadkomma detta utan datorstöd är ej möjligt. För att säkra spårbarhet mellan intressenter, behovsunderlag, behov och krav används verktyg från exempelvis QFD.

4.2 Modell för beskrivning av krav

Data och information som genereras i kravhanteringsprocessen måste kunna lagras i någon form av databas. Därför tas en datamodell fram som beskriver kravrelaterad data och information som genereras i kravhanteringsprocessen. Syftet är att skapa en generell applicerbar modell för att beskriva kravrelaterad data. Modellen ska användas som kärnan i ett datorstöd för kravhanteringsprocessen. Utvecklingen av modellen sker i tre steg, *Definition av modellen*, *Implementation av modellen* och *Utvärdering av modellen*. Dessa steg itereras för att evolutionärt förbättra modellen.

Definition av modell

Modell för krav definieras och beskrivs i UML.

Implementation av modellen

Modellen implementeras i en databas.

Utvärdering av modellen

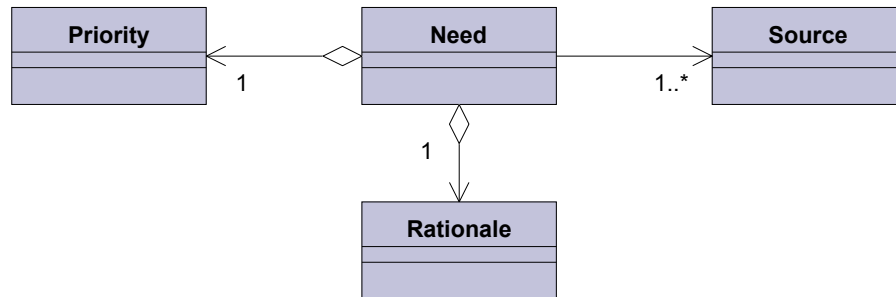
För att utvärdera modellen appliceras den på en fallstudie. Resultat av utvärderingen påtalar förbättringsförslag på modellen.

4.2.1 Teoretisk utgångspunkt

Utgångspunkten för datamodellen introduceras i detta stycke som en grundläggande klassmodell för beskrivning av relationerna mellan kravkällor, behov och slutgiltiga krav.

Behov

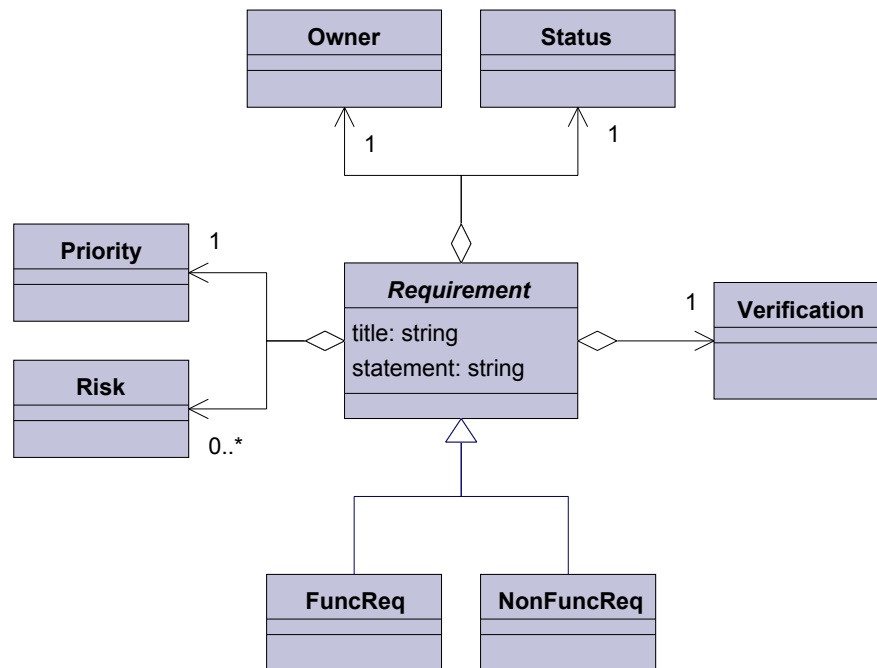
Under datainsamlingen samlas önskemål, restriktioner och kravställande dokument in som behovsunderlag. Detta tillsammans med vem eller vad som är upphov till dem samlas i begreppet källa (Figur 2). Filttrade önskemål bildar behov som representeras av motsvarande klasselement. Varje behov har en eller flera källor (intressent och kravställande dokument). Varje behov prioriteras och får en prioritet. Till varje behov anges dessutom en motivering av orsaken till dess uppkomst.



Figur 2 Klassdiagram beskrivande klassen *Need* (Behov).

Krav

Krav modelleras enligt UML-diagrammet i Figur 3. Den abstrakta basklassen *Requirement* symboliserar alla krav. Varje krav har en titel och en kravformulering. Varje krav har dessutom en prioritet, ett antal riskklassificeringar, en status, en ansvarig ägare samt en verifieringsmetod kopplat till sig. De fem sistnämnda kravattributen kan variera mellan olika projekts syfte och förutsättningar. Utgångspunkten för modelleringen i detta skede är dock att passa krav ställda på FMA. Prioriteten anger kravets grad av viktighet. Risk-klassificeringarna beskriver på vilka sätt kravet är kritiskt för projektet och dess genomförande. Kravets status kan anges som en uppräkningsstyp med värdena identifierat, granskat, implementerat, verifierat, validerat och avfört. Kravets ägare är ansvarig för att följa upp att kravet granskas, implementeras, verifieras och valideras. Verifieringsmetod beskriver hur kravet ska verifieras. Krav kan antingen vara funktionella eller icke-funktionella. Ett funktionellt krav beskriver något som systemet ska kunna utföra medan ett icke-funktionellt krav beskriver en begränsning på systemet.



Figur 3 Klassdiagram beskrivande klassen *Requirement* (Krav).

4.2.2 Datorstöd för kravhantering

För att kravhanteringen ska vara praktiskt hanterbar måste den stödjas av ett datorstöd. Detta behöver kravsificeras, designas, implementeras och utvärderas. Syftet är att utveckla ett datorstöd som stöder en kravhanteringsprocess och som baseras på datamodellen. Det är viktigt att datorstödet underlättar dynamik i kravhanteringsprocessen. Detta innebär exempelvis att ett nytt identifierat behov direkt skall reflekteras i motsvarande förändringar i kravbilden eller att ändringar av ställda krav direkt spåras till de identifierade behoven och visualiserar vad förändringen har för konsekvenser för uppfyllandet av de senare. Datorstödet implementeras som moduler där modulerna utgör datorstöd för metoder och tekniker. Modulerna skall kunna införas, bytas ut och modifieras oberoende av varandra. Datorstödet utvecklas i fyra steg som itereras integrerat med utvecklingen av processen och datamodellen.

Kravsificering

En kravlista på vad datorstöd för kravhantering skall klara av sammanställs med utgångspunkt i beskrivningen av kravhanteringsprocessen.

Design

En design görs av datorstödet. Designen bygger till hög grad på de metoder och tekniker för kravhantering som valts att implementeras.

Implementering

Datorstödet implementeras i exempelvis Microsoft Windowsmiljö utnyttjande Microsoft .Net-plattformen och programmeringsspråket C# med databasstöd från databashanteraren SQL Server.

Utvärdering

Flera utvärderingar av datorstödet genomförs, både i sin helhet och av de moduler som innefattar stöd till enskilda metoder och tekniker. Resultatet av utvärderingarna används som grund för utvecklingen av så väl datorstödet som kravhanteringsprocessen och datamodellen.

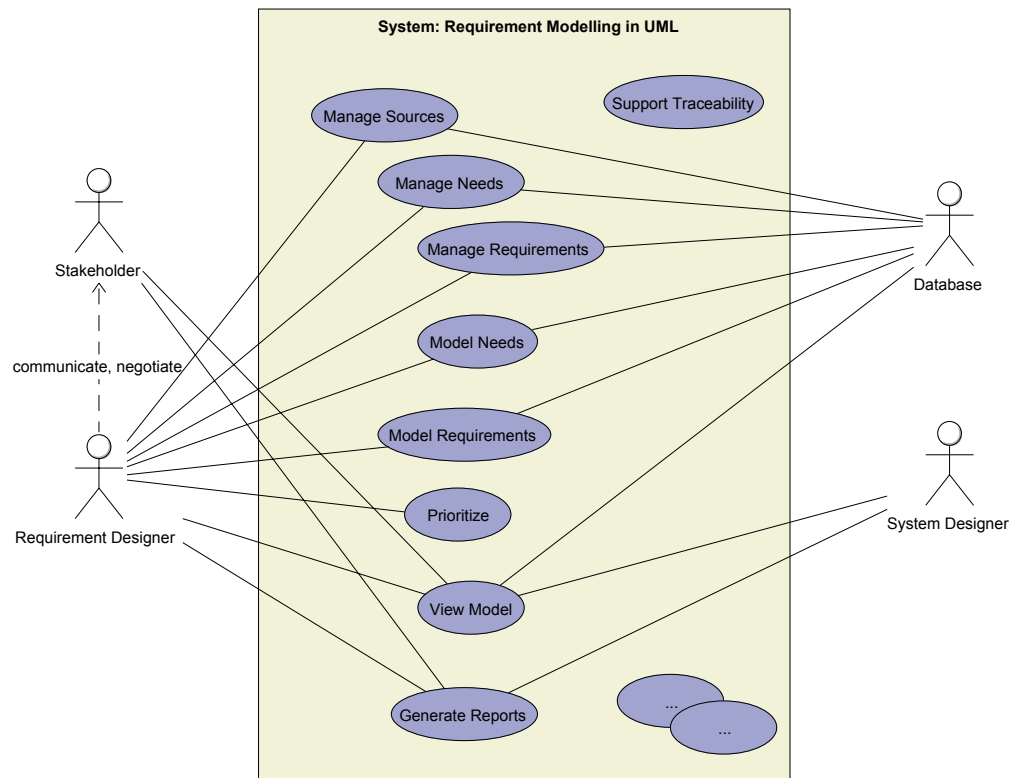
4.2.3 Teoretisk utgångspunkt

Utgångspunkten för utvecklingen av datorstödet för kravhantering beskrivs i detta stycke. Svårigheten att överblicka kravmängden ökar snabbt med storleken på systemet. Ett tusen detaljerade krav på ett mindre system är ingen ovanlighet. Traditionellt samlas behov och krav i en databas och klassificeras efter kravtyp och/eller tillämpningsområde. Att sätta sig in i kravmängden innebär en stor, om inte oöverkomlig, arbetsinsats.

Datorstödet skall:

- Stödja identifieringen av relevanta intressenter.
- Stödja framtagandet och analysen av behovsunderlag, behov och krav.
- Underlätta överblick av det framtida systemet
- Underlätta identifiering av saknade krav och överlappande krav.
- Underlätta för nya systemutvecklare att sätta sig in i befintlig kravmängd.
- Underlätta kommunikation med intressenter och systemutvecklare.

Det tänkta datorstödet funktionalitet illustreras i Figur 4. Funktionaliteten visas som ett användningsfallsdiagram och de följande beskrivningarna till respektive användningsfall redovisar enbart riktlinjer för tänkt funktionalitet.



Figur 4 Datorstöddets funktionalitet.

Hantera källor

Källor, d v s intressenter och kravställande dokument, ska kunna läggas in i verktyget. De ska kunna refereras från/till behoven och spåras från/till kraven. Källorna ska lagras i en databas. Källorna ska kunna ändras och tas bort.

Hantera behov

Behov ska kunna läggas till och lagras i databasen. Härledda krav ska kunna spåras från behoven och på samma sätt skall ett krav veta vilket behov det är härlett från. Behov ska kunna ändras och tas bort. Till varje behov ska dess källor kunna refereras, en prioritet kunna sättas samt en logisk beskrivning om varför det uppkommit kunna ges.

Hantera krav

Krav ska kunna läggas till och lagras i databasen. Krav är antingen funktionella eller ickefunktionella. Krav ska kunna ändras och tas bort. Verktøget ska stödja att krav kan fördes med klassificeringar vilka bestäms från projekt till projekt. Exempel på klassificeringar är:

- Kravtyp (prestanda, säkerhet, interna gränssnitt, gruppering m a p funktionalitet, m m).
- Ägare (vem som ansvarar för att kravet granskas, implementeras, verifieras, m m).

- Status (identifierat, granskat, implementerat, verifierat, validerat, förkastat, m m).
- Prioritet.
- Risker (finansiella, tekniska, personella, m m).
- Kostnad och tidsåtgång.
- Versionshantering (datum för införande samt senast modifierad).

Varje krav ska dessutom kunna förses med beskrivningar för hur det ska verifieras.

Modellera behov

Behov ska grafiskt kunna modelleras med hjälp av behovsdiagram (användnings- fallsdiagram). Till varje användningsfallselement refereras ett befintligt inlagt behov eller så införs behovet i samband med skapandet av användningsfallselementet. Behovsmodelleringen stöds även av aktörselement och klasselement. Varje användningsfallselement refererar ett kravdiagram (aktivitetsdiagram). Det ska vara möjligt att visualisera de olika diagrammen via elementen.

Modellera krav

Varje kravdiagram kan innehålla det grupperande elementet aktivitet och/eller övriga tillståndselement. Varje aktivitetselement refererar ett underliggande kravdiagram. Det ska vara möjligt att visualisera/traversera de olika diagrammen via elementen, både uppåt och nedåt i diagramsherakin. Till varje infört tillståndselement refereras befintliga krav eller så införs kraven i samband med skapandet av ett nytt tillståndselement.

Prioritera

Verktyget ska innehålla stöd för prioritering av behov och krav. Det finns ett flertal olika sätt att prioritera. Enklaste form av prioritering är att använda någon typ av skala, exempelvis högt, medel, lågt eller 1-5 för att värdera vikten av enskilda behov. Ett betydligt mer exakt, men mer resurskrävanden sätt att prioritera är parvisa jämförelser. Till exempel genom att *Behov A* är 5 gånger viktigare än *Behov B*.

Visualisera modell

Ett av syftena med verktyget är att underlätta för icke insatta att sätta sig in i mängden av behov och krav. Designen ska stödja navigering mellan de olika diagrammen, elementen, behoven, kraven och källorna. Krav och behov ska kunna hittas via elementen, element ska kunna hittas via krav och behov. Via grupperingselement ska motsvarande mängd krav kunna nås. Utskriftsmöjlighet ska tillhandahållas.

Generera rapporter

Verktyget ska ge stöd för generering av kravdokument av hela eller delar av underlaget. Val av delinnehåll ska kunna göras utifrån modellelement eller diagram.

Understöd spårbarhet

Verktyget ska ge stöd på spårbarhet genom att varje behov refereras med sina källor, varje krav har referenser till de behov det är härlett från. Källor i form av kravställande dokument kan dessutom lagras via verktyget i databasen.

4.3 Process för analys av krav

För att analysera hur befintliga arkitekturkrav motsvarar verksamheters behov av stöd från arkitekturen för att kunna genomföras så kan scenarion användas. I dessa scenarion används prototyper som utformats i det verksamhetsnära utvecklingsarbetet som en länk mellan användare och arkitekturen. Prototyper och scenarion blir därmed både en metod för att identifiera och validera krav samt ett sätt att säkra aktiv användarmedverkan i kravhanteringsprocessen. Scenarion bildar en utgångspunkt för diskussion kring arkitekturen och de krav som ställs på den. I scenarier kan användare både genom ord och uppträdande validera befintliga krav och påvisa behovet av ytterligare krav.

Syftet är att utveckla en process som stödjer identifiering och validering av arkitekturkrav. Denna process bygger på användandet av scenarion och prototyper. Utvecklingen av denna process sker i två steg som itereras.

Definition av process

Processen och de innefattade aktiviteterna definieras och utvecklas.

Utvärdering av processen

För att utvärdera processen appliceras den på en fallstudie, där data om utfallet samlas in med stöd av deltagande observationer. Resultatet av utvärderingen som påvisar förbättringsförslag av process används som underlag för att införa förbättringar.

4.4 Teoretisk utgångspunkt

Den teoretiska utgångspunkten för utvecklingen av processen för att analysera krav med stöd av scenarion och prototyper utgör även ett praktiskt underlag för en sådan process.

Process för utvärdering av arkitekturkrav med scenarier stödda av prototyper:

1. Identifiera ett antal lämpliga scenarion och prototyper
2. Under ett antal sessioner, kör scenariona med stöd av prototypen.
Dokumentera sessionen med video (ljud och bild)
3. En moderatorledd expertpanel studerar videodokumentation, med syfte att
 - a. Identifiera de sekvenser av scenarierna som är relevanta för arkitekturen
 - b. Identifiera de krav dessa ställer på arkitekturen
4. Jämför identifierade krav med de befintliga arkitekturkraven för att:
 - a. Validera befintliga krav
 - b. Identifiera krav som borde tas med i kravmängden för FMA

- c. Identifiera attribut och förtydliganden till redan dokumenterade FMA-krav

En utvärderingsstudie av processen för att analysera och validera krav genomförs således och processen uppdateras med utgångspunkt från detta. Resultatet av appliceringen av processen förs in i arbetet med arkitekturkraven.

5 Benämningar, akronymer och förkortningar

Aktör	En aktör är en roll som någon eller något har som interagerar med ett system, t ex en människa, en hårdvara eller ett annat system.
Användningsfall	Ett användningsfall specificerar en sekvens av aktioner, inklusive varianter, som ett system kan utföra. Användningsfallet ska generera ett observerbart resultat värdefullt för en speciell aktör.
Behov	Andra utvecklingsstadiet för ett krav. Behov är filtrerade och analyserade ur behovsunderlaget. Behov dokumenteras som användningsfall.
Behovsunderlag	Första utvecklingsstadiet för ett krav. Behovsunderlag är information insamlade om intressenternas direkt och indirekt formulerade krav på systemet, ofta i form av önskemål. Behovsunderlag dokumenteras skriftligen på det sätt de inhämtats utan att några omformuleringar görs. Dokumenten benäms kravställande dokument.
Funktionellt krav	Ett funktionellt krav beskriver vad ett system ska kunna göra.
Icke-funktionellt krav	Ett icke-funktionellt krav anger restriktioner på hur de funktionella kraven ska implementeras.
Intressent	En part som har en rättighet, del eller fordran i en enhet av systemet, eller i dess innehav av karakteristika, som möter denna parts behov och förväntningar (ISO/IEC 15 288, 2001).
Krav	Tredje och avslutande utvecklingsstadiet för ett krav. Krav extraheras ur behov. Krav ska omfatta behovens utsträckning. Krav dokumenteras på så sätt att de är verifierbara, kompletta, förståeliga, entydiga samt icke nedbrytbara. Den totala kravmängden får inte innehålla konflikter, överlappningar, utelämnningar eller inkonsistenser.
Kravställande dokument	Ett dokument innehållande behovsunderlag.

6 Referenser

- Braithwaite, T. (1993) Information Service Excellence Through TQM: Building Partnerships for Business Process Reengineering and Continuous Improvement, Milwaukee: ASQC Quality Press.
- Brooks, F. P. Jr. (1995) The Mythical Man-month: Essays on Software Engineering, Addison-Wesley.
- Buckingham Shum, S. (1996) Design Argumentation as Design Rationale, The Encyclopedia of Computer Science and Technology, Vol. 35, No. 20, s. 95-128.
- Carlshamre, P. (2001) A Usability Perspective on Requirements Engineering, Dissertation No. 626, Institute of Technology, Linköping University, Linköping
- Davenport, T. (1993) Process Innovation–Reengineering Work through Information Technology, Harvard Business School Press, Boston, MA.
- Ditsa, G. E. M. & Davies, J. (2000) Activity Theory as a Theoretical Foundation for Information Systems Research, in Proceedings of the International Conference on Information Technology Management in the 21st Century, Idea Group Publishing, Hersey, PA, pp. 240-244.
- Dorfman, M. (1996) Requirements Engineering, in Tayer, R. H. & Dorfman, M. (Eds.) Software Requirements Engineering. Second Edition, IEEE Computer Society Press, Los Alamitos, CA, s. 6-22.
- Graham, I. (1998) Requirements Engineering and Rapid Development. An Object Oriented Approach, Addison Wesley, Harlow.
- ISO/IEC 15288, System Engineering – System Life Cycle Processes, ISO/IEC CD 15288 FCDIS, Version 4, Material under bearbetning.
- Jacobson, I., Booch, G. & Rumbaugh, J. (1998) The Unified Software Development Process, Addison Wesley, Reading Mass.
- Kotonya, G. & Sommerville, I. (1998) Requirements Engineering. Processes and Techniques, John Wiley & Sons, Chichester.
- Kuutti, K. (1999) Activity Theory, Work, and Systems Design, i Engeström, Y., Miettinen, R. & Punamöki, R. (Eds.) Perspectives on Activity Theory, Cambridge University Press, Cambridge, s. 360-366.
- Morin, M. (2002) Multimedia Representations of Distributed Tactical Operations, Linköping Studies in Science and Technology, Thesis No. 661.
- Mumford, E. (1995) Effective Systems Design and Requirements Analysis. The ETHICS Approach, Macmillan Press Ltd.

Persson, P. A. (2000) Bringing Power and Knowledge Together. Information Systems Design for Autonomy and Control in Command Work, Linköping Studies in Science and Technology, Thesis No. 639.

Robertson, S. & Robertson, J. (1999) Mastering the Requirements Process, Addison-Wesley.

Schuler, D. & Namioka, A. (1993) Participatory Design. Principles and Practices, Lawrence Earlbaum, Hillsdale, NJ.

Siddiqi, J. & Shekaran, M. C. (1996) Requirements Engineering: the Emerging Wisdom, IEEE Software, Vol. 13, No. 2, s. 15-18.

Sommerville, I. & Sawyer, P. (1996) Requirements Engineering. A Good Practice Guide, John Wiley & Sons, Chichester.

Young, R. (2001) Effective Requirements Engineering. Addison-Wesley.