

Att skapa beslutsstödssystem från existerande system genom att använda Soft Systems Methodology och Prototyping

Magisteruppsats i Informatik
Handelshögskolan vid Göteborgs Universitet
Institutionen för Informatik

Författare: Tommy Axelsson
Maria Hansson
Niklas Kjellander

Handledare: Kjell Engberg

Examinator: Urban Nuldén

Abstrakt

Uppsatsen syftar till att undersöka om det är möjligt att utveckla beslutsstödssystem, med utgångspunkt från ett antal existerande system, med hjälp av vår sammanställda utvecklingsmetod. Utvecklingsmetoden utgår ifrån *Soft Systems Methodology* och *Evolutionär Prototyping*. Undersökningen baseras på en fallstudie utförd på *Ericsson Mobile Data Design AB* (ERV) vars resultat bygger på deltagande observationer av utvecklarna, intervjuer med användare och beställare, samt kvalitetstestning av systemet. Utvecklarna upplevde att kombinationen av *Soft Systems Methodology* och *Evolutionär Prototyping* bidragit till framtagandet av beslutsstödssystemet. Användare och beställare uppskattade kombinationen av struktur och anpassningsbarhet i utvecklingsmetoden. Kvalitetstesterna påvisade att systemet uppfyllde användarnas behov. Vår slutsats är att den sammanställda utvecklingsmetoden gick att använda i den beskrivna fallstudien, för att utveckla ett beslutsstödssystem av existerande system. Vi anser att utvecklingsmetoden går att använda i liknande fall där syftet är att skapa ett beslutsstödssystem av ett antal existerande system.

INNEHÅLLSFÖRTECKNING

1 INLEDNING	1
1.1 BESLUTSSTÖDSSYSTEM.....	2
1.2 BESLUTSSTÖDSSYSTEMETS DELAR OCH KONSTRUKTION.....	4
1.3 UTVECKLINGSMETOD.....	8
1.4 PROBLEMFÖRMULERING.....	14
1.5 AVGRÄNSNINGAR.....	14
2 FORSKNINGSMETOD	15
2.1 VETENSKAPSTEORETISKT STÄLLNINGSTAGANDE.....	15
2.2 KVALITATIV ELLER KVANTITATIV METOD.....	17
2.3 UNDERSÖKNINGSSTRATEGI.....	21
3 RESULTAT	29
3.1 UTVECKLARNAS SAMMANSTÄLLDA DAGBÖCKER.....	29
3.2 UTVÄRDERING UTVECKLINGARBETE ENLIGT UPPDRAGSGIVARNA.....	35
3.3 UTVÄRDERING AV KVALITET ENLIGT KVALITETSMALL.....	38
4 DISKUSSION	47
4.1 HAR DEN BESKRIVNA UTVECKLINGSMETODEN ANVÄNTS PÅ ETT KORREKT SÄTT?.....	47
4.2 MOTSVARAR SYSTEMET ERVS BEHOV?.....	49
4.3 FINNS DET YTTRE FAKTORER SOM PÅVERKAT UTVECKLINGSMETODENS RESULTAT?.....	51
4.4 VIDARE FORSKNING.....	51
KÄLLFÖRTECKNING:	52
BILAGOR	55

FIGURFÖRTECKNING

FIGUR 1 LANGEFORS INFOLOGISKA EKVATION. LANGEFORS, 1995	1
FIGUR 2 GORRY & SCOTT-MORTON'S KLASSIFICERINGSMATRIS. TURBAN OCH ARONSON, 1995.	2
FIGUR 3 KLASSIFICERING AV MODELLER. FIGUREN BASERAS PÅ TURBAN OCH ARONSON, 1995.	5
FIGUR 4 TVÅ AV ANVÄNDARGRÄNSSNITTETS TRE DELAR. TURBAN OCH ARONSON, 1995.	5
FIGUR 5 BESLUTSSTÖDSSYSTEM SOM SKAL TILL EXISTERANDE SYSTEM. FIGUREN BASERAS PÅ SPRAGUE OCH WATSON, 1993.	6
FIGUR 6 CHECKLANDS TVÅ VÄRLDAR: VERKLIGA VÄRLDEN OCH DEN TÄNKTA VÄRLDEN OM DEN RIKTIGA VÄRLDEN (LEWIS, 1994).	9
FIGUR 7. INTERN OCH EXTERN KVALITET MED SEX HUVUDEGENSKAPER OCH 27 UNDEREGENSKAPER ENLIGT ISO 9126-1 (HÅKANSSON, 2000.).	23
FIGUR 8 KVALITET VID ANVÄNDNING INNEFATTAR EN HUVUDEGENSKAP OCH FYRA UNDEREGENSKAPER ENLIGT ISO 9126-1 (HÅKANSSON, 2000).	23
FIGUR 9 ARKITEKTUR ÖVER PROTOTYPEN.....	33
FIGUR 10 RELATIONER MELLAN DELARNA I TESTSPECIFIKATIONEN.	38
FIGUR 11 DEL AV TESTSPECIFIKATIONEN: EGENSKAPEN FUNKTIONALITET	38
FIGUR 12 RESULTAT FRÅN TESTNING AV FUNKTIONALITET	40
FIGUR 13 DEL AV TESTSPECIFIKATIONEN: EGENSKAPEN TILLFÖRLITLIGHET	41
FIGUR 14 RESULTAT FRÅN TESTNING AV TILLFÖRLITLIGHET	41
FIGUR 15 DEL AV TESTSPECIFIKATIONEN: EGENSKAPEN ANVÄNDBARHET	42
FIGUR 16 RESULTAT FRÅN TESTNING AV ANVÄNDBARHET	43
FIGUR 17 DEL AV TESTSPECIFIKATIONEN: EGENSKAPEN PRODUKTIVITET	43
FIGUR 18 RESULTAT FRÅN TESTNING AV PRODUKTIVITET	44
FIGUR 19 DEL AV TESTSPECIFIKATIONEN: EGENSKAPEN UNDERHÅLLSMÄSSIGHET.....	44
FIGUR 20 RESULTAT FRÅN TESTNING AV UNDERHÅLLSMÄSSIGHET.....	44
FIGUR 21 DEL AV TESTSPECIFIKATIONEN: EGENSKAPEN FLYTTBARHET	45
FIGUR 22 RESULTAT FRÅN TESTNINGEN AV FLYTTBARHET	45
FIGUR 23 DEL AV TESTSPECIFIKATIONEN: EGENSKAPEN ANVÄNDARKVALITET.....	45
FIGUR 24 RESULTAT FRÅN TESTNING AV ANVÄNDARKVALITET	46

1 Inledning

Ericsson Mobile Data Design AB (ERV) är ett företag i Ericsson-koncernen. En del av ERV utvecklar och säljer MOBITEX-nät (Ericsson AB, 2001) som används för trådlös paketbaserad datakommunikation. En operatör är en individ eller ett företag som ansvarar för driften av ett MOBITEX-nät. Problemet för operatören är att han måste söka information från ett flertal system för att kunna fatta beslut rörande driften av MOBITEX-nätet. ERV är intresserade av att underlätta driften av nätet.

Ett MOBITEX-nät består i huvudsak av en *central* (NCC) och ett antal *basstationer* (BASAR). Användarna av nätet, *mobiler*, kommunicerar genom att skicka data via basstationerna.

Operatören kan hämta information om mobiler och basstationer genom att använda sig av nätets *underhållssystem* (OM). OM tillhandahåller viss momentan information om mobiler, men sparar inte informationen för vidare bearbetning. Information om trafiken under en basstation lagras i *loggfiler* som går att undersöka med en texteditor. Vid felsökning av nätet är det användbart att se hur mycket trafik som går via en viss basstation. Information om det här finns i loggfilerna. OMs nackdel är att det är kommandobaserat i stordatormiljö. Det här innebär att det ställs stora krav på operatören som dels måste vara insatt i MOBITEX, dels vara insatt i de operativsystem som används.

Det som ERV söker är dels ett gemensamt systemgränssnitt mot de olika systemen för att underlätta kommunikationen, dels grafiska användargränssnitt som underlättar tolkning av data från de olika källorna (se bilaga 1).

Den beskrivna situationen präglas av ett flertal informationskällor, vilka är mer eller mindre tillgängliga. Problemet med alltför många informationskällor är att det blir svårt att skapa sig en överblick och att lyfta ut den relevanta informationen i en specifik situation, för att få ett optimalt beslutsunderlag. Det här kan liknas vid *informationsstress* (Lotsson 1998). Enligt den infologiska ekvationen (Langefors, 1995) är information för individen beroende av tolkningen av data (D) under en viss tid (t) samt individens begreppsvärld, verklighetsuppfattning och värderingar (S) (se Figur 1). För att skapa information av data behöver individen tid samt egna referensramar. Kan ett system öka överblickbarheten och koncentrera sig på relevanta data, d.v.s. öka meningsfullheten, medför det att individen inte behöver lika mycket tid för att skapa sin tolkning. Det är den här tolkningen som utgör grundval för det aktuella beslutet.

$$I = i(D, S, t)$$

*I = information, i = tolkningsprocess, D = data, t = tid
S = begreppsvärld, verklighetsuppfattning och värderingar*

Figur 1 Langefors infologiska ekvation. Langefors, 1995

Ett led i att öka överblickbarheten är att minska kontaktytan mellan användare och de system som existerar, att endast ge tillgång till den funktionalitet och information som är väsentlig. De olika systemen kan ses som att de är inkapslade i en "svart låda" där användaren endast ser indata och tillhörande utdata, vilket oftast är det som användaren är intresserad av (Brown, 1997).

Med den här tekniken behöver inte användaren veta var data kommer ifrån och hur den görs tillgänglig, fokus hamnar på vilken indata som finns och vad systemet genererar för utdata. Det här leder till att mer energi kan läggas på beslutsfattande, och inte insamlingen av underliggande data. Ett sätt att åstadkomma det här är att skapa ett beslutsstödssystem som hämtar data från existerande system.

1.1 Beslutsstödssystem

Beslut är något som berör alla dagligen både privat och professionellt. Beslutsfattande av god kvalitet är tidskrävande och komplicerat. Mycket tid spenderas alltså på att fatta beslut i olika situationer. Informationsflödet omkring oss påverkar besluts kvaliteten och den som kan hantera det på rätt sätt fattar således bättre beslut. Beslutsstödssystem hjälper till att hantera stora mängder med information och plockar fram det som är väsentligt för att fatta bra beslut. (Turban och Aronson, 1995.)

De största fördelarna med beslutsstödssystem är enligt Turban och Aronson (1995):

- ?? Snabba beräkningar
- ?? Utökar människans kognitiva förmåga
- ?? Kostnadsreduceringar
- ?? Tekniskt hjälpmedel
- ?? Förbättrad kvalitet

Det finns en mängd olika beslutsstödssystem med specifika egenskaper. Gorry & Scott-Morton (Turban och Aronson, 1995) har gjort en klassificeringstabell för beslutsstödssystem, vilket hjälper oss att välja rätt sorts system utifrån de behov som finns. Deras arbete kan sammanfattas i Figur 2.

Typ av beslut	Typ av planering			Typ av system för att täcka behov
	Operationell planering	Taktisk planering	Strategisk Planering	
Strukturerat	Ekonomisk rapportering, order rapportering	Budget analysis, short-term forecasting, personal reports, make or buy	Financial management (investment), warehouse location, distribution systems	Management information system, operations research models, transaction processing
Semi strukturerat	Planering av produktion, inventering	Kreditutvärdering, budgetarbete, produktionslayout, schemaläggning av produktion, belöningsystem	Bygga en ny anläggning, sammanslagning och uppköp, planera för en ny produkt, planera för kompensering, kvalitetssäkring	Decision Support Systems
Ostrukturerat	Välja förstasida till en tidning, köpa mjukvara, godkänna lån	Förhandling, rekrytera chefer, köpa hårdvara, lobbyism	Forskning och utveckling, utveckla ny teknologi, planera socialt ansvar	Decision Support Systems, Expert Systems, Neural networks
Typ av system för att täcka behov	Management information system, management science	Management science, Decision Support Systems, Expert Systems, Executive Information System	Executive Information System, Expert Systems, neural networks	

Figur 2 Gorry & Scott-Morton's klassificeringsmatris. Turban och Aronson, 1995.

Gorry och Scott-Mortons matris bygger på Simon och Anthonys teorier. Simon anser att arbete kan delas in efter hur väl strukturerat utförandet är, vilket motsvaras av de vågräta raderna. I vertikalled bygger matrisen på Anthonys teorier som ser till vilken typ av beslut som systemet skall stödja. Han delar in beslut som operationella, taktiska och strategiska. Operationella beslut rör det dagliga arbetet i produktionen. De taktiska besluten handlar om hur organisationens resurser skall utnyttjas. De strategiska besluten syftar till att sätta upp de mål som skall gälla för hela organisationen.

Olika typer av beslutssituationer kräver olika typer av beslutsstöd, men det går att ta det här resonemanget ytterligare ett steg. Är det så att beslutsfattande har olika behov under beslutsprocessen? Enligt Sprague och Watson (1993) består beslutsprocessen av fyra aktiviteter, informationsinsamling, utformning, val och utförande, vilka kan gynnas av olika typer av beslutsstödssystem. Turban och Aronson menar att vissa typer av system är mer lämpade än andra för valda delar av beslutsprocessen.

En faktor som gör beslutsstödssystem speciella är att beslutsfattare har en s.k. kognitiv stil, ett personligt sätt som de samlar och sorterar information. Det här ställer i sin tur krav på att beslutsstödssystem är flexibla och inte alltför generella beträffande t.ex. presentationen av information, då alla har sina personliga preferenser på hur de vill att information skall projiceras. Slutligen påverkas valet av beslutsstödssystem av antalet beslutsfattare som är inblandade i ett beslut. (Turban och Aronson, 1995.)

Vår bedömning av ERVs situation pekar ut DSS och MIS som möjliga lösningar på problemet. Det här grundar vi på att systemet är tänkt att användas av en användare, d.v.s. det är inte frågan om ett system för beslutsfattande i grupp. Vidare har vi identifierat informationsinsamling som ett av de primära användningsområdena.

1.1.1 Decision Support System

”**Decision Support System (DSS)** Computer-based information system that combine models and data in an attempt to solve non-structured problems with extensive user involvement.” (Turban och Aronson, 1995, s. 856.)

Ett DSS stödjer beslutsfattaren genom att ställa analytiska resurser till förfogande. Det här förutsätter att det finns viss struktur i det problem som skall lösas. Beslutsfattarens roll är att bidra med sitt omdöme. Fördelen med ett DSS är att beslutsfattarens förmåga att analysera stora mängder information förbättras, varvid effektiviteten ökar. Beslutsfattarna uppskattar DSS för dess stödjande effekt och att det inte är till för att automatisera beslutsprocessen, eller komma med förslag till lösningar. DSS används bland annat till att skapa underlag för kreditupplysningar och förberedande budgetarbete. (Turban och Aronson, 1995.)

1.1.2 Management Information System

”**Management Information System (MIS)** A business information system designed to provide past, present and future information appropriate for planning, organising and controlling the operations of the organisation.” (Turban och Aronson, 1995, s. 863.)

MIS används huvudsakligen i strukturerade problemsituationer där besluten följer standardiserade regler och informationsflödet går att förutsäga. Ett MIS största fördelar är att det ökar effektiviteten, minskar kostnader, minskar genomloppstider och ersätter kontorspersonal. Beslutsfattare påverkas indirekt av ett MIS, då det ger

tillgång till rapporter och data. Ett typiskt användningsområde för MIS är bokföringssystem. (Turban och Aronson, 1995.)

1.1.3 Vårt ställningstagande

Ovanstående teorier talar för att ERV skulle gagnas av ett beslutsstödssystem. Enligt Simons klassificering (se Figur 2) anser vi att DSS är den typ av system som bäst svarar mot behoven och den rådande situationen. De arbetsuppgifter som systemet skall stödja är av semistrukturerad typ. Det här kommer vi fram till genom att utesluta de båda andra kategorierna. Arbetsuppgiften kräver allt för mycket erfarenhet och intuition av beslutsfattaren för att uppgiften skall kunna ses som väl strukturerad. Vi anser även att arbetsuppgifterna är tillräckligt strukturerade för att inte kunna klassificeras som ostrukturerade, då det finns vissa mönster och som återkommer och specifika data som är intressant att mäta.

Ur Anthonys (se Figur 2) perspektiv är ERV i behov av ett DSS eller MIS då de beslut som systemet skall stödja är av både taktisk och operationell karaktär. De taktiska besluten handlar om att utnyttja och omfördela de resurser som finns i MOBITEX-nätet. Besluten på operationell nivå handlar om att övervaka den dagliga driften av nätet.

Enligt Finlays (1994) definitioner av beslutsstödssystem är ERV i behov av ett MIS, då låg osäkerhet präglar svaren som systemet förväntas leverera. Eftersom systemet skall hantera och analysera data från förfluten tid pekar Finlays definition på att systemet skall vara ett *Data Retrieval System* (DRS). Finlay delar in DRS i tre underkategorier där ERV skulle ha mest nytta av ett *Data Analysis System* (DAS), vilket omfattar analyser, rapporter och grafiska presentationer av förutbestämda situationer.

Det är svårt att placera ett system i ett fack, d.v.s. att det är antingen det ena, DSS, eller det andra, MIS. Enligt resonemanget finns det saker som pekar på att ERV skulle behöva ett DSS, men det finns även sådant som tyder på att de skulle få användning av ett MIS. Ett problem vid klassificering av beslutsstödssystem är den tvetydighet och brist på tydliga definitioner som råder. Finlay (1994) visar på det här problemet genom att illustrera hur olika författare ser på klassificeringen av MIS och DSS. På grundval av det här väljer vi Gorry och Scott-Mortons klassificering (se Figur 2), varför vi fortsättningsvis kommer att referera till vårt beslutsstödssystem som ett DSS, då vi anser att det är den term som över lag svarar bäst mot ERVs behov.

1.2 Beslutsstödssystemets delar och konstruktion

1.2.1 Modeller

Modellbasen är en del av kärnan i ett beslutsstödssystem som bidrar till dess karaktäristiska egenskaper. Det är modellbasen som skiljer ett beslutsstödssystem från ett vanligt informationssystem. Turban och Aronson (1995) delar in modeller i två huvudkategorier, statiska och dynamiska. En statisk modell kan liknas vid ett fotografi, en beskrivning av något vid en bestämd tidpunkt. Statika modeller förutsätter att all relevant data är stabil, d.v.s. oföränderlig. Dynamiska modeller beskriver en företeelse under en viss tidsperiod. Det intressanta med en dynamisk modell är att belysa de förändringar som inträffar, vilket underlättar vid identifiering av trender och regelbundna mönster i data.

Modeller skapas med hänsyn till om de skall verka under förhållande som präglas av säkerhet eller osäkerhet, vilket resulterar i att det finns fyra grundtyper av modeller (se Figur 3). En modell byggs upp av ett antal mindre beståndsdelar som syftar till att uppfylla modellens helhetsegenskaper. Enligt Turban och Aronson (1995) består en modell av en eller flera av följande delar: heuristiska, simulerande, statistiska, finansiella samt optimerande.

	Statisk	Dynamisk
Säkerhet	modell	modell
Osäkerhet	modell	modell

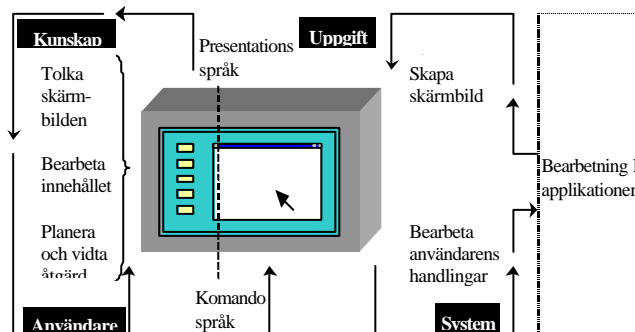
Figur 3 Klassificering av modeller. Figuren baseras på Turban och Aronson, 1995.

Ett beslutsstödssystem inkluderar en eller flera typer av modeller beroende på vad systemet skall användas till. En bra modell kännetecknas av att den är enkel och överskådlig och samtidigt uppfyller de krav som är satta för systemet som helhet. Enligt Brown (1997) definieras en modell som en förenklad avbildning av en del av verkligheten.

1.2.2 Användargränssnitt

Mycket av systemets kraftfullhet, flexibilitet och användbarhet är relaterat till dess användargränssnitt. Det är den här delen av systemet som användaren ser, vilket medför att användargränssnittets egenskaper i stor grad påverkar hur användaren uppfattar systemet. Användargränssnittet är ett delsystem, vilket enligt Sprague och Watson (1993) kan delas upp i användare, terminal och mjukvara. De definierar dialogen mellan användare och system i tre delar.

1. Kommandospråk – avser vad användaren kan göra rent fysiskt för att kommunicera med systemet. Dit räknas t.ex. nyckelord, funktionstangenter, touchpaneler, mus etc.
2. Presentationsspråk – är det som användaren ser. Hit räknas alternativ som skrivare, bildskärm, grafik, färger, ljud etc.
3. Kunskap – innebär i det här avseendet vad användaren måste veta för att använda systemet på ett effektivt och meningsfullt sätt. Kunskapen kan finnas i användarens huvud, i en användarmanual, i ett instruktionsblad eller som inbyggda hjälpfunktioner i systemet.



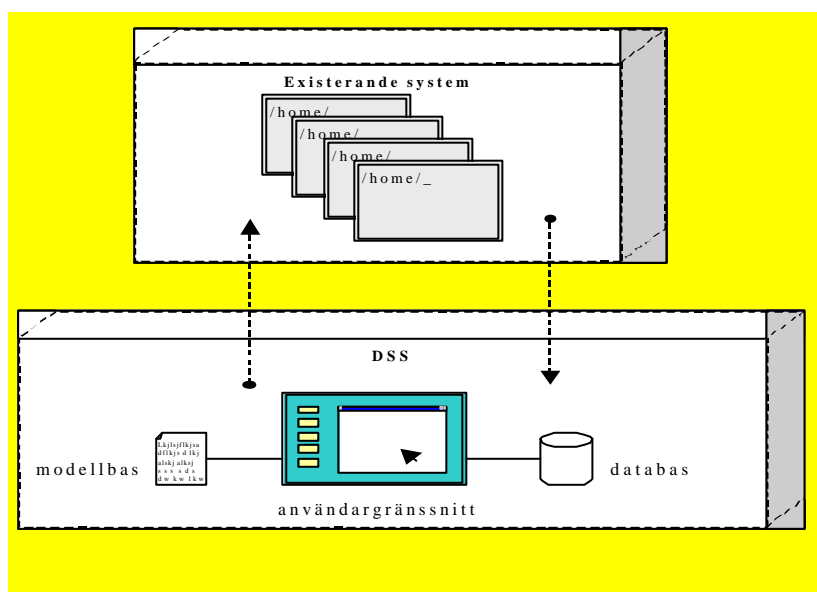
Figur 4 Två av användargränssnittets tre delar. Turban och Aronson, 1995.

Hur bra användargränssnittet upplevs är beroende på var och en av de tre delarna. En annan aspekt av hur delsystemet upplevs beror vilken form av dialog som används. De vanligaste typerna av dialog är text, menyer, frågor och svar, blankettifyllnad och grafiska användargränssnitt (GUI). Ett bra användargränssnitt är flexibelt så att det går att alternera mellan olika dialogformer. Att gränssnittet går att anpassa efter användarens önskemål och erfarenhet bidrar också till den upplevda kvaliteten ökar. För att ett beslutsstödssystem skall uppfattas som bra måste det kunna presentera data i olika format och media, samt att det finns hjälp tillgänglig om hur de olika dialogerna skall användas. (Turban och Aronson, 1995)

1.2.3 DSS som ett skal

Som vi nämnt så har ERV ett antal existerande system som förser operatören med information. Idag är samtliga av dem textbaserade och körs på en stordator. De saknar alltså grafiska gränssnitt likväl som de saknar funktionalitet för att bearbeta och projicera data ur olika perspektiv.

Ett alternativ är att bygga ett system från grunden, vilket medför mycket extra arbete. Det här är helt onödigt då mycket av arbetet redan är gjort. De existerande systemen behöver inte ersättas, utan kan istället tjäna som datakälla för beslutsstödssystemet. Beslutsstödssystem består i princip av ett grafiskt användargränssnitt, en modellbas och en databas för att spara data i. Det här är den grundläggande arkitektur för hur ett beslutsstödssystem är uppbyggt, vilken beskrivs av Sprague och Watson (1993) (se Figur 5).



Figur 5 Beslutsstödssystem som skal till existerande system. Figuren baseras på Sprague och Watson, 1993.

Med det grafiska användargränssnittet kommunicerar användaren med de underliggande systemen som tidigare var tillgängliga genom deras respektive textbaserade gränssnitt. I modellbasen finns regler som är relevanta för den data som skall samlas in och presenteras för att stödja beslutsfattandet. Databasen gör det möjligt att spara data för att kunna titta på ett beslutsunderlag vid ett senare tillfälle, samt att lagra den data som ingår i beslutsunderlagen. På det här viset uppnås effekten av en "svart låda" som diskuterades i inledningen till kapitel 1.

1.2.4 Konstruktion av DSS

Konstruktionen av ett DSS ser olika ut beroende på vilken typ och kategori av DSS det är. Det beror även mycket på vilken sorts organisation beslutsstödssystemet skall införas i.

1.2.4.1 Utvecklingsstrategier

Det finns olika strategier för att utveckla DSS. På 80-talet var det vanligt att utveckla kundspecifika DSS i universalverktyg som Cobol eller Pascal. De här systemen var oftast mycket storskaliga. Idag är det vanligare att använda *fjärde generationens språk* (4GL) och dataorienterade språk, vilket ökar produktiviteten betydligt. Att använda integrerade utvecklingsverktyg för DSS kan öka effektiviteten. Om systemet skall innehålla flertalet olika funktioner är det effektivare att använda integrerade paket än att sammanställa flera individuella moduler i 4GL. Utveckling av DSS kan även göras genom CASE-metodologi. Vid utvecklandet av ett komplext DSS är det vanligt att integrera flera av de nämnda strategierna. (Turban och Aronson, 1995.)

1.2.4.2 Utvecklingsprocessen

I utvecklingsprocessen för ett DSS ingår, enligt Turban och Aronson (1995):

1. **Planering**, i denna fas ses till vilka behov som finns och vilka problem är som skall lösas. De beslut som beslutsstödssystemet skall stödja definieras.
2. **Undersökning**, innebär att resurserna identifieras och hur utvecklaren på bästa möjliga sätt uppnår planeringens resultat.
3. **Systemanalys och konceptuell design**, här fastställs det lämpligaste konstruktionsförfarandet och de resurser som krävs för att implementera den.
4. **Design**, under denna fas specificeras systemet komponenter, struktur och egenskaper.
5. **Konstruktion**, hur systemet konstrueras beror på designfilosofi och de verktyg som skall användas, konstruktionen är realiseringen av designen. När systemet konstrueras testas det kontinuerligt och förbättras.
6. **Implementation**, den här fasen består av följande moment, där många ofta utförs samtidigt:
 - ?? Testning, utdata jämförs med designspecifikationen.
 - ?? Utvärdering, här utvärderas hur väl systemet stämmer överens med användarnas behov.
 - ?? Demonstration, en demonstration av det färdiga systemet för användarna är mycket viktigt för att de skall acceptera systemet.
 - ?? Användarinstruktioner, här instrueras användarna i systemets grundläggande möjligheter och funktioner.
 - ?? Träning, användare får lära sig hur systemet är uppbyggt och hur det skall underhållas.
 - ?? Användning, systemet kan användas.
7. **Underhåll och dokumentation**, innebär planering för support av systemet och användarna. Dokumentation för användning och underhåll av systemet skrivs.
8. **Anpassning**, efter hand behöver systemet anpassas efter nya behov som användaren får. Det här görs genom att gå igenom de tidigare stegen kontinuerligt.

Det finns flera sätt att utveckla ett DSS på, där de två ytterligheterna är livscykelmodellen och evolutionär prototyping. Vid utformandet av ett DSS är det krångliga att kartlägga beslutsprocesserna. Det är svårt för utvecklarna att förstå dessa

processer, speciellt då det är vanligt att användarna inte själva är medvetna om hur de fattar beslut. Många utvecklare har därför valt att gå ifrån den traditionella livscykelmodellen och istället övergått till en speciell typ av prototyping, evolutionär prototyping. Evolutionär prototyping går ut på att bygga DSS i ett antal mindre steg med direkt respons från användaren. De verktyg som används för att utveckla DSS måste därför medföra att det går att göra förändringar på ett snabbt och enkelt sätt. (Turban och Aronson, 1995.)

1.2.5 Framgångsfaktorer för konstruktion av ett DSS

För att ett beslutsstödssystem skall bli framgångsrikt menar Finlay (1994) att användarmedverkan är avgörande. Under konstruktionen av ett beslutsstödssystem skall därför följande punkter beaktas:

- ?? Avdela mer tid för att ta reda på vad användarna verkligen vill ha.
- ?? Var uppmärksam på vilka kunskaper användarna har och hur de fattar beslut.
- ?? Behåll ett nära samarbete under hela utvecklingsarbetet.
- ?? Förse användarna med en service snarare än en produkt.
- ?? Utveckla ett enkelt system som inte överbelastar användarna.
- ?? Använd en evolutionär metod som utnyttjar användarnas inläring.

Finlay (1994) påstår även att hur användarmedverkan skall tillämpas är beroende av vilken typ av beslutsstödssystem som skall konstrueras. För att ett utvecklingsprojekt av ett DSS skall lyckas så är projektstyrning viktigare än hårdvara, mjukvara och teknisk expertis. Finlay (1994) säger att bra projektstyrning bygger på:

- ?? En tydlig plan för hela utvecklingscykeln.
- ?? Omsorgsfull ansvarsfördelning inom projektet.
- ?? Lämplig användarmedverkan och styrning.
- ?? Fortlöpande utredning av användarens behov.

1.3 Utvecklingsmetod

För att utveckla ett DSS används sedvanliga metoder för systemutveckling, men med fokus på olika delar beroende på vilken typ av DSS det är. I avsaknad av specifik metod för att utveckla olika typer av beslutsstödssystem är det upp till utvecklaren att designa ett lämpligt tillvägagångssätt av olika metoder för den specifika situationen, vilket vi har valt att göra med hjälp av *Soft System Methodology* och *Prototyping*.

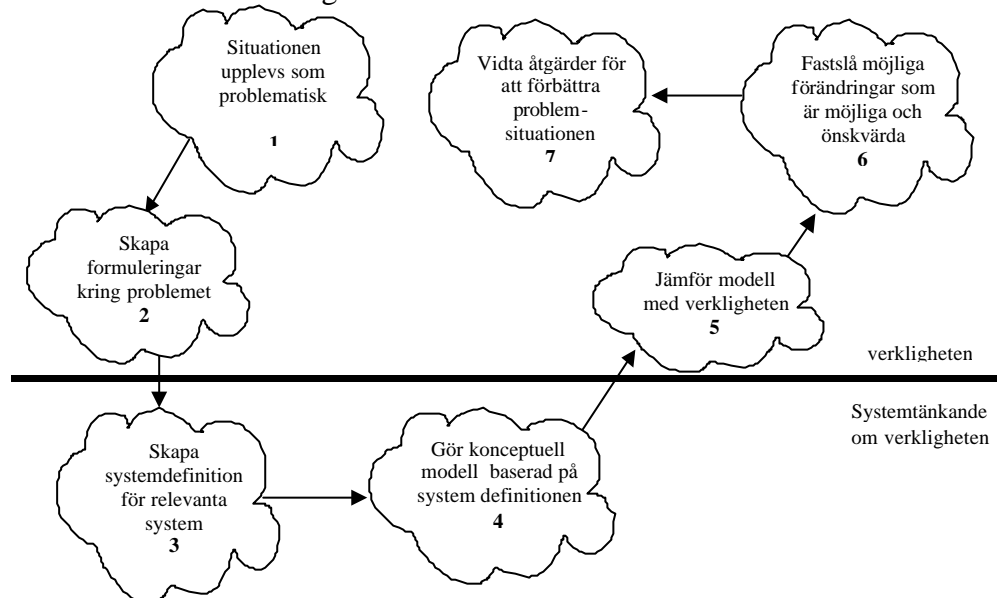
1.3.1 Soft Systems Methodology

Problematiken med utveckling av beslutsstödssystem beskrivs bl.a. av Thomas Finne (1998). DSS används ofta som ett mellanliggande lager vilket förmedlar kunskaper utvecklaren har haft om problemområdet. Därför måste många källor användas och utvärderas för att få ett pålitligt resultat.

Själva grundprincipen med *Soft Systems Methodology* är att den angriper problematiken med dåligt definierade, "röriga", problem i den verkliga världen (Lewis, 1994). SSM består av sju steg, som visas i Figur 6. Hur noggrant varje steg behöver utföras beror till stor del på det problem som skall lösas. Stegen behöver inte användas i den angivna ordningen, vissa steg kan till och med exkluderas om

situationen så kräver. Det tillvägagångssätt vi beskriver här är därför endast ett av många sätt som SSM kan utföras på.

I SSM ingår det att utreda problemet ur två olika synvinklar, dels den logiska utredningen, dels den kulturella utredningen. Aktiviteterna i de sju stegen (se Figur 6) skall medföra ett system som uppfyller kraven samtidigt som det passar in kulturellt i organisationen. Utvecklaren måste studera den sociala verkligheten som han verkar i och förändrar. Utvecklaren behöver även vara medveten om att han påverkar processen genom att studera den. Det är viktigt att förstå de politiska processerna för att förstå de bakomliggande orsakerna till problemet, men även för att komma till insikt med varför ett initiativ togs till att lösa det.



Figur 6 Checklands Två världar: Verkliga världen och den tänkta världen om den riktiga världen (Lewis, 1994).

Under det första steget, *situationen upplevs som problematisk*, måste utvecklaren, för att kunna skapa sig en uppfattning om de sociala och politiska förhållandena, klargöra vissa roller. De roller som skall klargöras härrör dels från problemsituationen där ingripandet sker, dels från själva ingripandet. När utvecklaren skall sätta sig in i problemsituationen måste han beakta att det tar tid att passa in i organisationen, vilket leder till ökad förståelse för problematiken. Det tar även mycket tid att inse vilka grupper som finns och vilket inflytande de har i organisationen.

För att utvecklaren skall bilda sig en uppfattning om problematiken, alltså kunna *skapa formuleringar kring problemet*, finns två huvudsakliga linjer att arbeta efter, *rika bilder* och *undersökning*. *Rika bilder* används i första hand för att utvecklare, användare och beställare skall kunna bilda sig en gemensam uppfattning om problemsituationen. En annan möjlighet är *undersökning* av problematiken, vilket främst går ut på att hitta strukturer i problemsituationen och därigenom identifiera vilka processer som ingår.

Eftersom det är osäkert när övergången sker från de huvudsakligen utredande stegen i ett och två till de mer modellerande stegen tre och fyra (se Figur 6), menar Checkland (Lewis, 1994) att det inte är avgörande att ha den kontrollen. Efter att utvecklarna har bildat sig en grundläggande förståelse för problematiken, har de tillräcklig kunskap

för att kunna bilda sig en uppfattning om vilka olika delsystem som måste konstrueras. Resultatet är inga speciellt detaljerade modellbeskrivningar men de tjänar sitt syfte som utgångspunkt för det fortsatta arbetet. Under denna fas är det i huvudsak tre beskrivningar som måste göras för att kunna fortsätta processen. De är *omvälvande eller konservativ modellering, systemdefinition med hjälp av CATWOE och aktivitetsmodeller*.

Efter de inledande stegen finns en formell beskrivning av systemet, men det kan vara av nytta att åstadkomma en mer detaljerad beskrivning av systemet, d.v.s. *en konceptuell modell baserad på systemdefinitionen*. Det här åstadkoms genom att utveckla för varje aktivitetsmodell ställer sig frågan: Hur kan ett system beskrivas som utför en sådan aktivitet? Alltså beskrivs de olika delsystemen mer detaljerat för att öka förståelsen för helhetsegenskaperna hos systemet.

Finlay (1994) har vidareutvecklat Checklands modell. Steg fyra har expanderats i två delar, 4a och 4b, där det antingen går att följa Checklands rekommendationer för att skapa en konceptuell modell eller ersätta den med en extern metod.

Under de avslutande stegen, fem, sex och sju, verifieras systemet med användare och beställares verklighet. Eventuella avvikelser bedöms efter hur allvarliga de uppfattas, och beslut om åtgärder fattas.

1.3.2 Prototyping

Ett DSS är till sin natur mycket annorlunda jämfört med andra system, därav krävs det ett annorlunda angreppssätt för att designa och konstruera ett sådant system. Det traditionella tillvägagångssättet med analys, design och implementering har visat sig vara otillräckligt då det inte finns någon uttömmande teori som förklarar beslut och beslutsfattande. De teorier som finns är otillräckliga då beslutsfattande påverkas av personliga referensramar och beslutsfattarens omgivning som är under ständig förändring. Systemdesigners har således svårt att beskriva i teorin vad systemet skall omfatta, då användare och beslutsfattare inte kan precisera vad ett beslutsstödssystem skall komma att omfatta. (Turban och Aronson, 1995.)

En annan orsak till att det är svårt att ha en enhetlig metod för att utveckla beslutsstödssystem är att det finns så många varianter, allt från välstrukturerade till helt ostrukturerade system. Turban och Aronson (1995) rekommenderar att ett DSS konstrueras i korta steg med kontinuerlig respons från användarna, för att konfirmera att utvecklingen sker i rätt riktning. Sprague och Watson (1993) föreslår att beslutsstödssystem skall byggas med s.k. iterativ design som tillåter att systemet lätt och snabbt kan förändras. En utvidgning av det här förslaget är den utvecklingsmodell som presenteras av Turban och Aronson (1995). Gemensamt för Sprague/Watson och Turban/Aronsons metoder är att de inleds med en förstudie som resulterar i en kravspecifikation. När kravspecifikationen är klar fortsätter den iterativa designen tills systemet är färdigt. Turban och Aronsons (1995) modell ger utvecklaren en valmöjlighet för att styra hur utvecklingen skall bedrivas efter det att kravspecifikationen är klar. Den ena vägen innebär att systemet utvecklas enligt en livscykelmodell. Den andra möjligheten är att systemet utvecklas genom evolutionär prototyping. Även om det är frågan om att utveckla ett system för att stödja beslut i ett välstrukturerat problemområde så kan det vara svårt att genomföra en traditionell analys och design då användaren inte känner till sina behov eller är helt införstådd i

på vilka grunder de fattar sina beslut. Det är alltså möjligt att använda en metod som bygger på en livscykelmodell, men utvecklare verkar föredra att använda sig utav evolutionär prototyping istället (Turban och Aronson, 1995). Den här formen av prototyping syftar till att, genom en serie av korta utvecklingssteg, bygga ett system med fortlöpande respons från användaren. Utvecklingsarbetet kan delas upp i fyra aktiviteter:

1. Välj ut ett delsystem att börja med. Användare och utvecklare utser en del av systemet där utvecklingen skall starta. Interaktion med användarna på ett tidigt stadium borgar för god fortsatt kommunikation. Delsystemet skall inte vara större än att det går att överblicka situationen. En god idé är att börja med något som beslutsfattaren är intresserad av för att skapa gott engagemang.
2. Utveckla en liten men användbar version av systemet. Ingen direkt analys utföres för att hålla uppe hastigheten i utvecklingsprocessen.
3. Utvärdera det färdiga delsystemet. Efter varje cykel skall systemet utvärderas i samarbete med användarna. Utvärderingen används som kontroll för att hålla projektet på rätt spår. I slutet av varje utvärderingsfas tas beslut om fortsatt utveckling eller om systemet kan anses vara färdigt.
4. Förfina, utöka och modifiera systemet i cykler.

De här aktiviteterna upprepas tills dess att systemet är färdigt. Den data som behövs för systemet specificeras under arbetets gång, vilket är en av de stora fördelarna med metoden. Så länge systemet bara skall ge stöd åt en beslutsfattare fungerar metoden väl, men när det är frågan om ett system för flera användare, beslutsfattande i grupp, blir det mer komplicerat och mer resurser måste avsättas för att skapa en god kommunikation mellan utvecklare och användare. De största fördelarna med att använda evolutionär prototyping vid utveckling av ett DSS, enligt Turban och Aronson (1995), är:

- ?? Kort utvecklingstid.
- ?? Snabb respons från användarna.
- ?? Användarna får ökad förståelse för systemet, dess informationsbehov och dess möjligheter.
- ?? Lägre utvecklingskostnader.

1.3.3 Sammanställd utvecklingsmetod

När vi stod inför valet av utvecklingsmetod för att kunna skapa ett DSS av ett antal existerande system såg vi två möjliga kandidater, SSM och prototyping. Vi konstaterade även att ingen av dem innehöll exakt det vi var ute efter.

Styrkan med SSM är dess förmåga att reda ut ostrukturerade situationer och vagt definierade problem. Ofta kan kunden inte specificera vad som är fel, utan snarare att allt inte riktigt är som det ska. SSM börjar här med att ta reda på vem som är problemägaren. Tillsammans med den försöker utvecklaren skapa en bild över situationen med hjälp av rika bilder och slutligen en systemdefinition, där systemets grundläggande krav är definierade. Det finns nu en någorlunda klar bild av vad som skall göras, varför och var det är lämpligt att börja. Som vi tidigare har förklarat vet problemägaren inte hur hela systemet kommer att se ut, då han eller hon inte har en bra uppfattning om på vilka grunder och hur de fattar sina beslut. Det är alltså svårt att skapa en konceptuell modell av systemet genom att endast använda formella metoder.

Styrkan med prototyping är att det går att prova sig fram och utveckla systemet bit för bit, allt eftersom nya fakta blir kända. Den typen av prototyping kallas evolutionär prototyping, vilken efter hand skapar en komplett konceptuell modell som stämmer väl överens med det kunden vill ha. Nackdelen med prototyping är att det inte ingår någon metod för att belysa det grundläggande problemet innan konceptualiseringen inleds.

Vi ser därför många fördelar i att kombinera SSM och evolutionär prototyping. Det vi ser som prototypings stora svaghet, i det här sammanhanget, ser vi samtidigt som SSMs styrka, och vice versa. SSM används således för att strukturera och definiera problemet, och prototyping för att konceptualisera problemet.

Innehållsmässigt är SSM utformad för att vara generell d.v.s. det skall gå att tillämpa den på så många områden som möjligt. Det är upp till användaren att bestämma vilka av stegen i SSM som skall användas och i vilken ordning. Det går även att utföra stegen på ett personligt sätt. Vår sammanställda utvecklingsmetod är därför vår egna variant av SSM där vi introducerar evolutionär prototyping i konceptualiseringen.

För att utveckla ett DSS av ett antal existerande system har vi utgått från de sju stegen i SSM i den ordning som Checkland föreslår och vi har namngivet dem enligt följande:

1. Planering
2. Förstudie
3. Definition av system
4. Konceptualisering
5. Utvärdering
6. Fastslå förändring
7. Beslut

Planeringsfasen startar i och med att beställaren inser att det finns ett problem i organisationen och söker efter en lösning. För att komma fram till det övergripande målet med projektet samt yttre tidsramar skall utvecklaren ha en inledande diskussion med beställaren. Med yttre tidsramar menas att de viktigaste händelserna i projektet sätts i ett tidsperspektiv, t.ex. i en tidsplan. Vid utveckling av DSS är det svårt att skapa sig en komplett bild av problemområdet då det är krångligt att kartlägga beslutsprocesser. Utvecklaren skall därför undvika att i den här fasen endast se problemsituationen ur ett perspektiv.

Under förstudien är syftet att vidareutveckla det övergripande mål som planeringsfasen leder fram till och vidare kartlägga de centrala besluten som systemet skall hantera. Förstudien skall utreda hur situationen skall förändras genom att dels se till den rådande situationen, dels till hur den förbättrade situationen skall se ut. En förutsättning för att utvecklaren skall kunna skapa sig en bild över den rådande situationen är att de identifierar rollerna beställare och användare. Utvecklaren skall sedan utforska vilka de upplevda symptomen är för respektive roll, då de har olika syn på problemområdet. Det är även viktigt att utreda vilka resurser som finns, vilka de existerande system är och hur beslut har fattats tidigare. Information om det här går att samla in genom observationer, tidigare dokumentation men framför allt genom att

diskutera med de involverade. Slutsatser om hur situationen skall förbättras dras från den insamlade informationen tillsammans med det övergripande målet. Ett lämpligt sätt att dokumentera och presentera den nuvarande situationen samt ett eller flera förslag till förändring är rika bilder (Lewis, 1994). Med rika bilder uppnås överblick över systemet.

De rika bilderna, eller motsvarande, ligger till grund för att skapa en mer precis beskrivning av systemet. Varje rik bild ger upphov till en eller flera systemdefinitioner. För att få med de nödvändiga delarna i definitionen kan utvecklaren med fördel använda sig av en vedertagen valideringsmetod såsom CATWOE (Lewis, 1994) eller VATOFF (Mathiassen, 1998). En sådan metod klarlägger vilka de inblandade är och hur de påverkas av systemet. En viktig faktor vid valet av metod är på vilket sätt de skiljer sig åt i framtoningen. CATWOE inriktar sig mer på de sociala aspekterna av systemutveckling medan VATOFF är mer inriktad på tekniken som skall användas.

Utvecklaren skall specificera problemområdet ytterligare genom att kartlägga de centrala processerna. Det är även här som tekniken och omgivningen blir en del av problembilden. Valideringsmetoden skall även beskriva vilken funktionalitet systemet skall erbjuda. Kunden väljer en systemdefinition utifrån vilken en kravspecifikation tas fram. Kravspecifikationen skall förutom systemdefinitionen innehålla en mer detaljerad beskrivning av de krav som gäller systemet funktionalitet. Kravspecifikationen fungerar som ett kontrakt mellan utvecklare och beställare (Sommerville, 2001). För att säkerställa att kraven uppfylls så skall en testspecifikation utformas utifrån kravspecifikationen. Den underlättar även för utvecklaren då han med hjälp av den vet mer exakt vilka mål han förväntas uppfylla.

Förutsättningen för att börja konceptualisera är att kravspecifikationen är godkänd. För att konceptualisera systemet används evolutionär prototyping. Börja med den del av systemet som användaren är särskilt intresserad av. Utveckla en liten och användbar version av systemet. Utvärdera versionen och gör eventuella förändringar för att sedan utöka systemet med en ytterligare del. Processen fortlöper tills dess att användare och utvecklare anser att kraven är uppfyllda.

Prototypen skall nu utvärderas utefter testspecifikationen för att kontrollera att den uppfyller kravspecifikationen. Om systemet inte uppfyller alla krav återupptas konceptualiseringen för att korrigera de brister som systemet uppvisat.

När systemet uppfyller testspecifikationens krav utvärderar beställare och användare om det finns ytterligare krav som skall utöka kravspecifikationen. De eventuella kraven skall specificeras vart efter både kravspecifikation och testspecifikation revideras.

I beslutsfasen bestäms om de nya kraven skall konceptualiseras. Om kraven skall genomföras återgår projektet till konceptualiseringsfasen. I annat fall är systemet klart för implementering.

1.4 Problemformulering

En lösning på ERVs problem och andra liknande situationer är att samla flera applikationer under ett gemensamt skal i form av ett DSS. För att utveckla det gemensamma skalet, så att det hjälper användaren till bättre beslutsfattande, har vi sammanställt en utvecklingsmetod.

Uppsatsen fokuserar på att undersöka om utvecklingsmetoden är till nytta för ERV och utvecklare i liknande situationer då det är aktuellt att skapa ett DSS av ett antal existerande system. För att kunna avgöra om utvecklingsmetoden har gett rätt resultat skall det tydligt framgå att det framtagna systemet underlättar insamling, bearbetning och presentation av data för att förbättra besluts kvaliteten. Enligt infologiska ekvationen (se Figur 1) förbättras tolkningsprocessen (i) då data (D) presenteras på ett för ändamålet bättre sätt samt att onödiga data filtreras bort. För att nå dit måste utvecklingsmetoden ta hänsyn till användarnas begreppsvärld, värderingar och verklighetsuppfattning (S). Vi vill alltså undersöka om utvecklingsmetoden bidrar på ett väsentligt sätt till framtagandet av ett DSS samt vad metodens olika delarna bidrar med.

Det här ger upphov till följande frågeställning:

Kan den beskrivna utvecklingsmetoden användas för att skapa ett DSS av ett antal existerande system?

1.5 Avgränsningar

I vår uppsats vill vi undersöka en utvecklingsmetod för att skapa ett DSS av ett flertal existerande system. Vår undersökning kommer att begränsa sig till en studie på ERV. På grund av de yttre tidsgränser som råder för vårt arbete kommer vi endast att utföra ett varv av iterationsprocessen i vår utvecklingsmetod. Det här medför att stegen "fastslå förändring" och "beslut" förlorar sin betydelse för vår studie och uppsats, då dessa steg endast avgör om det färdiga systemet skall vidareutvecklas. Vi avgränsar oss därför till att undersöka de fem första faserna i vår utvecklingsmetod. Det här medför även att det inte går att mäta och utvärdera effekterna av det färdiga systemet.

2 Forskningsmetod

2.1 Vetenskapsteoretiskt ställningstagande

2.1.1 Frågeställningar och val

Innan en vetenskaplig undersökning påbörjas, finns det ett antal viktiga frågor att ta ställning till beträffande utredningens upplägg. De flesta av de här valen kan verka självklara, men för att säkerställa metodens korrekthet och fullständighet rekommenderar Easterby-Smith (1991) att forskare tar ställning till följande fem frågor.

- ?? Forskarens inblandning.
- ?? Urvalets storlek.
- ?? Typer av teorier.
- ?? Typ av studie.
- ?? Verifikation eller falsifikation.

Det första ställningstagandet behandlar forskarens inblandning i det material som skall undersökas. Forskningshistoriskt sett skall forskaren se det insamlade materialet utifrån, s.k. "utifrån-in perspektiv", men i dagsläget är det vedertaget att aktivt delta i undersökningsmaterialet.

Den andra ställningstagandet en forskare måste göra är att ta hänsyn till storleken på populationen som skall undersökas. Forskaren kan antingen välja att samla data från en stor fluktuerande population eller att fokusera på en liten homogen population. Det första fallet syftar till att belysa relationerna mellan olika populationer och hur vissa faktorer kan variera inom en population. Nyckelfrågan är att bestämma hur stor en testpopulation måste vara för att vara representativ. Det andra fallet används för att studera hur populationer förändras över tiden i vissa avseenden. Här är problemet att identifiera de relevanta förändringsprocesserna och att studera dem under en tillräckligt lång tidsperiod.

Det tredje ställningstagandet berör vad som skall komma först, data eller teorier. Glasser och Strauss har formulerat ett angreppssätt för att utveckla teorier, känt som "the grounded theory" (Easterby-Smith, 1991). Det här angreppssättet fokuserar på insamlandet av data rörande specifika händelser eller processer i olika miljöer, för att skapa en grund för kommande teorier. Den viktigaste uppgiften för forskare, enligt "the grounded theory", är att utveckla teorier enligt denna jämförande metod. Det motsatta förfaringsättet är att starta med en teori, eller en hypotes, för att sedan utreda om data stödjer eller motsäger teorin. Hypotestestning har en praktiskt fördel på grund av sina måldrivna egenskaper.

Den fjärde frågan handlar om forskningen skall baseras på experimentell design eller fältarbete. Experimentell design går ut på att genomföra tester på utvalda grupper, och studera resultatet. Fördelen med experimentell design är att det är en intuitiv metod. Nackdelen med den här metoden är att den är svår att genomföra i situationer där försöket är beroende av människor, då det kan vara svårt att få tillgång till en testgrupp av organisatoriska skäl. Alternativet till experimentell design är fältarbete,

som innebär att organisatoriska och sociala förhållanden undersöks. Den vanligaste fältarbetsmetoden är etnografi, men även kvantitativa metoder kan appliceras. Fördelen är att forskaren på ett naturligt sätt får ett "inifrånperspektiv" på problemet, men det kan samtidigt vara svårt att uppnå det förtroende, från gruppen som undersöks, som är nödvändigt för att nå resultat.

Den sista frågan rör ställningstagandet mellan verifiering och falsifiering, vilka båda används för att genomföra induktionsbevis. Verifiering innebär att data samlas in för att, i första hand, bekräfta en tes medan falsifiering, i första hand, innebär att forskaren letar efter data som motsäger tesen.

2.1.2 Vårt ställningstagande

Ovanstående frågeställningar försöker belysa de ramar, inom vilka forskningen skall bedrivas. Oftast går det inte att välja en renodlad metod som endast grundar sig på ett av paradigmen, utan det blir en blandning av de båda som tillsammans utgör en situationsanpassad forskningsmetod. Samtliga frågeställningar färgas av ens egna personliga preferenser och den situation forskaren befinner sig i, varför vi enades om följande:

- ?? Inifrånperspektiv.
- ?? Liten undersökningspopulation.
- ?? Teori före data
- ?? Experimentell design.
- ?? Verifiering.

Undersökningspopulationen bestod av de personer som utvecklade systemet enligt utvecklingsmetoden samt beställare och användare av systemet. Det fanns endast ett begränsat antal beställare och användare som hade rätt kvalifikationer för att ingå i vår undersökning. Alternativet till det här var att utöka populationen genom att observera flera utvecklingsprojekt på ett eller flera ställen. Då vi inte hade tillräckliga resurser för att studera flera projekt begränsade vi oss till ett.

Vi undersökte hur väl en utvecklingsmetod fungerade i en specifik situation. Den data vi samlade in bekräftar eller motsäger vår frågeställning. Det som undersöktes var bl.a. de som använde den föreslagna utvecklingsmetoden för att utveckla ett DSS åt ERV. Vi var här intresserade av att få ett "inifrånperspektiv" på undersökningen. Ett lämpligt sätt som vi såg för att göra det här var att vi som forskare genomförde utvecklingsarbetet med det tilltänkta systemet.

Det vi ville undersöka var om metoden är lämplig för den givna typen av situation, d.v.s. vi hade för avsikt att genomföra en hypotestestning. Fördelen med hypotestestning är att det går relativt snabbt att genomföra undersökningen. Nackdelen var om det skulle visa sig att metoden inte var lämplig måste processen göras om från början, förutsatt att det fanns ett behov av en fungerande utvecklingsmetod.

Vi hade för avsikt att genomföra en experimentell design där ett systemutvecklingsprojekt låg till grund för utvärderingen av vår utvecklingsmetod. Resultatet baseras på det arbete som utvecklarna utfört, beställarens uppfattning om

vad som utförts och på vilket sätt det utförts, samt utvärderingen av det system som producerats. Vår undersökning bekräftades alltså genom verifiering.

2.2 Kvalitativ eller kvantitativ metod

För att samla in data finns det två olika huvudriktningar, kvalitativt och kvantitativt. Metodernas namn avspeglar omfattningen av den insamlade data, d.v.s. den kvalitativa metoden samlar in förhållandevis lite data jämfört med kvantitativa metoden som omfattar en större mängd insamlad data för att ge en rättvis bild av det som undersöks. Valet av metod är nära relaterat till vilken typ av undersökning som utförs och vilka restriktioner som existerar. (Easterby-Smith, 1991.)

2.2.1 Kvalitativa metoder

Att arbeta med kvalitativa metoder innebär ofta att forskaren arbetar nära med dem som skall undersökas. De vanligaste metoderna är intervjuer, observationer och dagbok. Gemensamt för dem är att de underlättar för forskaren att skapa en överblick på människor och situationer. Metoderna underlättar för de undersökta att uttrycka sig om sin omvärld och situation, vilket är första gången för de flesta. Valet av insamlingsmetod skall väl avspegla den forskningsmetod som används. (Easterby-Smith, 1991.)

2.2.1.1 Intervjuer

Denna metod anses ofta som den bästa av de kvalitativa metoderna, men den kan lätt bli alltför komplex. Att genomföra intervjuer är oerhört tidskrävande. Den enklaste formen av intervjuer kan nästan likställas med enkäter där varje fråga kan besvaras med ett kort svar. I de mer komplexa typerna av intervjuer kräver frågorna ingående svar och det hela ter sig mer som en konversation. Intervjuarens roll är att styra konversationen i rätt riktning. Ofta är det inte bara svaret som är av intresse utan även artikulering, röstläge eller vad en intervjuade har för kläder på sig. Syftet med intervjuer är att forskaren skall kunna förstå de intervjuades situation i ett visst avseende som forskaren inte kan förutse. Det krävs mycket träning för att bli duktig på att genomföra intervjuer, då det handlar om så mycket mer än att endast ställa frågor och teckna ner svaren. Ofta måste forskaren hjälpa de intervjuade att utveckla sina svar och att komma till vissa insikter. Hur strukturerad intervju forskaren väljer att göra beror på vilka erfarenheter vederbörande har av tidigare intervjuer. En mer erfaren utredare har lättare för att hålla s.k. öppna intervjuer, men aspekter som hur känd den undersökta data är gör det lättare att hålla den här typen av intervju. De viktigaste egenskaperna hos en intervjuare är att kunna sortera ut den relevanta informationen i svaren och spara den på ett lämpligt sätt. Andra viktiga egenskaper är att kunna styra intervjun och att hålla inne på sina egna åsikter och värderingar. (Easterby-Smith, 1991.)

2.2.1.2 Komplement till intervjuer

Utöver de svar som de intervjuade lämnar ifrån sig är det även intressant att veta lite mer om dem och vilka värderingar de har. Det kan t.ex. gå ut på att rangordna ett antal företeelser inbördes eller oberoende av varandra. Intervjuer måste inte ske med en person i taget, utan kan lika gärna omfatta en grupp. Gruppen kan komma fram till helt andra slutsatser än vad deltagarna hade gjort var för sig. Genom att de intervjuade kan argumentera med varandra kan de även enas eller komma till nya insikter i de aktuella frågorna, vilket kanske inte hade varit fallet om de blivit intervjuade i enrum. (Easterby-Smith, 1991.)

Ytterligare ett alternativ till att utföra intervjuer är enkäter vilka är särskilt lämpliga när det är frågan om frågeställningar som fodrar långa svar. Då undviks den s.k. intervjuareffekten samt att det går att behandla lite mer känsliga frågor då den utfrågade känner en högre grad av anonymitet. De stora nackdelarna är att det är svårt att få rätt svar första gången samt att det inte går att ställa följdfrågor då det inte säkert går att säga vem som påstår vad. Andra vanliga problem är låg svarsfrekvens och att insamlingen tar lång tid. (Eriksson, Wiedersheim-Paul, 1997.)

2.2.1.3 Observationer

Olika typer av metoder för observationer kommer från etnografen. Etnografen började som en metod för att förstå sig på stammar i främmande kulturer, vilka studerades under en längre tid. En organisation kan lätt liknas vid en stam med egna uttryck och normer. Därav är observationer ett användbart verktyg vid studier av en organisation. När observationer genomförs är det viktigt att klargöra vilken roll forskaren skall ha i den undersökta organisationen. En metod är att ingå i organisationen precis som vilken medarbetare som helst. Den här metoden är bra när forskaren i första hand vill bekanta sig med det dagliga arbetet. En av nackdelarna med den här typen av undersökningar är att den oftast måste hållas dold för de övriga medarbetarna. Får de reda på att de blir övervakade kan t.ex. "Hawthorne-effekter" uppstå och bidra till ett missvisande resultat. Att observera öppet har de fördelarna att forskaren slipper många etiska problem som kan uppstå då en dold undersökning genomförs. Nackdelarna är att det kan ta lång tid innan medarbetarnas förtroende vinnns. (Easterby-Smith, 1991.)

2.2.1.4 Dagbok

Den här metoden går ut på att låta den undersökta organisationen föra dagbok över allt de gör under en tid. Att föra dagbok kan mätas både kvalitativt och kvantitativt. En av de stora fördelarna är att det går att observera flera personer simultant. (Easterby-Smith, 1991.)

2.2.2 Kvantitativa metoder

Fastän kvantitativa metoder mest förknippas med stora vetenskapliga undersökningar, går det även att använda sig av de här metoder för att undersöka relativt små populationer. En av de stora fördelarna med kvantitativa metoder är att datainsamlingen är skild från analysen.

De 4 mest spridda kvantitativa metoderna för datainsamling är väl definierade av Easterby-Smith (1991):

- ?? **Intervjuer**, används ofta i marknadsundersökningar eller opinionsundersökningar. Intervjuaren har precist formulerade frågor och förväntar sig precisa svar, vilket kan styras genom utformningen av frågeformuläret.
- ?? **Tester och mätningar**, kan användas för att ta reda på individers uppfattningar. Oftast utförs de som en serie frågor (50-100 st.) med Ja/Nej-svar, där svarsmönster jämförs med normaliserade svar för att bedöma det undersökta.
- ?? **Observationer**, räknas oftast som en kvalitativ metod men det går även standardisera och systematisera observationerna för att skapa en kvantitativ metod. Den mest använda är *aktivitetsinsamling* och innebär att korta observationer av processer eller människor görs med regelbundna mellanrum.
- ?? **Frågeformulär**, används främst i storskaliga undersökningar av politiska eller konsument preferenser. Fastän deras resultat kan tyckas enkla att använda och analysera måste mycket kraft läggas på deras utformning.

2.2.3 Val av empirisk metod

Gällande förutsättningar och tidsramar för vårt undersökningsproblem talar för att vi skulle använda kvalitativa metoder för insamling av data. Vi anser att datainsamlingen inte var kvantifierbar då undersökningspopulationen var liten och vi endast kunde undersöka ett utvecklingsprojekt. De metoder som har använts är intervjuer och deltagande observationer.

Under konceptualiseringsfasen utförde vi deltagande observationer av oss själva som utvecklare. Målet med de deltagande observationerna var att få en omfattande bild av kvaliteten i utvecklingsarbetet, sett ur utvecklarens perspektiv. Den här delen av datainsamlingen gav oss information om utvecklingsmetodens styrkor och svagheter. Vilka delar av utvecklingsmetoden upplevde utvecklarna bidrog till att underlätta respektive försvåra utvecklingen av beslutsstödssystemet. Data om utvecklingsarbetet har insamlats genom att utvecklarna skrivit dagbok, som har sammanställts. Dagbok har förts varje dag, på papper, av samtliga utvecklare under de olika stegen av utvecklingsprocessen. Fördelen med att vi genomförde deltagande observationer i utvecklingsarbetet var att vi fick god tillgång till primärdata rörande den här delen av utvecklingsarbetet. Nackdelen var att vi påverkade resultatet då vi var medvetna om det som skulle undersökas.

För att få ett annat perspektiv på hur lämpad utvecklingsmetoden var för ändamålet lät vi beställaren och användaren, ERV, besvara ett antal frågor som ställde metoden i relation till resultatet, och hur vägen dit har upplevts. Det finns två sätt att samla in den här data: enkät eller intervju. Gemensamt för de här metoderna är att det finns ett klart syfte med varför de skall användas. Innan insamlandet kan börja måste forskaren veta vilken information som är intressant, vad informationen skall användas till och av vem. Det är även viktigt att forskaren bestämmer hur många som skall besvara undersökningen innan den konstrueras. När den önskade informationen insamlats är det viktigt att den granskas så att de som besvarat frågorna har tolkat dem korrekt.

Vi beslutade oss för att genomföra intervjuer med en beställare och en användare på ERV, vilka var de som kunde uttala sig om utvecklingsmetoden. På ERV fanns endast ett tiotal personer som var inblandade i driften av MOBITEX-nät. Syftet med

intervjuerna var att utreda beställare och användares syn på utvecklingsmetoden. Tyckte de att den höll god kvalitet enligt deras standard och påverkade metoden kommunikation, inflytande och samarbete mellan dels utvecklare och beställare, dels utvecklare och användare. Det här var information som det var svårt att säga något om ur ett mer objektivt perspektiv, d.v.s. att visa teoretiskt att en metod var bra för ett visst ändamål. Nackdelen var att de som använde utvecklingsmetoden kanske inte höll tillräckligt hög standard för att ge rättvisa åt metoden. Anledningen till att vi valde intervju framför enkät var att det är lättare att få in rätt data i första försöket, då det går att styra den typen av intervju och ställa följdfrågor. Fler fördelar var att mycket av det som upplevdes som negativt med intervjuer inte var något problem för oss då vi inte hade svårt att boka tid med de som skulle intervjuas, det krävdes inga långa resor samt att de frågor som skulle avhandlas inte var av känslig och personlig karaktär. Den svaghet med intervjuer som kvarstod var att det skulle kunna uppstå en intervjuareffekt, då vi saknade större erfarenhet av den här typen av datainsamling.

Då vi ansåg att vi hade begränsad erfarenhet i intervjuteknik såg vi det som mest lämpligt att arbeta utifrån en strukturerad intervjuguide (se bilaga 2). För att ta fram frågorna delade vi upp guiden i olika kategorier: kommunikation, inflytande, samarbete och kvalitet. För att komma fram till lämpliga frågor var vi tvungna att under varje kategori diskutera vad vi ville bekräfta. Det var viktigt att formulera frågorna så öppna som möjligt för att få fram perspektiv som var svåra att förutse. Eventuella följdfrågor under intervjun ställdes endast för att utveckla svar och inte för att belysa nya infallsvinklar. Vi intervjuade personerna var för sig för att motverka att de skulle påverka varandra. Enligt Easterby-Smith (1991) krävs det dessutom än mer av intervjuledaren vid intervju av grupp. Personerna som vi var intresserade av att intervjua hade stor inblandning i projektet och arbetade i nära kontakt med utvecklarna. Skapandet av intervjuguiden kom att följa nedanstående riktlinjer.

Syftet med kategorin kommunikation var att få ett underlag för utvärdering av hur beställaren upplevt kontakten med utvecklarna, samt om graden av kontakt hade skilt sig åt under de olika utvecklingsfaserna. Frågorna skulle även undersöka om beställaren känt sig tillräckligt informerad under utvecklingsarbetet, och om utvecklarnas kunskaper om tekniken hade upplevts tillräckliga för att kunna föra en fruktsam dialog.

För att utreda de intervjuades upplevda inflytande fokuserade frågorna på hur väl utvecklarna lyssnat till beställarens krav och önskemål. Hur beställaren uppfattat sitt inflytande var nära sammankopplat med hur väl samarbetet fungerat. För att belysa samarbetet ställde vi vårt arbete, som utvecklare, i relation till om systemet istället hade utvecklats av externa konsulter, respektive företagets egna utvecklare.

För att kunna utreda kvaliteten ställde vi frågor beträffande beställarens uppfattning om hur väl arbetet genomförts, och om det var rätt arbete som utförts. Vi ville även få fram en mer allmän bild av vad kvalitet innebar för beställaren och hur väl prototypen motsvarat den här bilden.

Frågorna under rubriken "Övrigt" i intervjuguiden berörde hur väl beställarna känt att systemet var definierat innan utvecklarna kom in i bilden. Intresset för det här grundade sig i att beslutsstödssystem ofta är svåra att definiera på förhand.

2.3 Undersökningsstrategi

2.3.1 Fallstudier

Grundstenarna i fallstudier består i att välja ut ett fåtal objekt och studera dem utifrån ett flertal perspektiv. Fallstudier har traditionellt använts för att t.ex. analysera beslutsprocesser i organisationer. Principen är att två antaganden görs, världen existerar och dess existens går att upptäcka. Fallstudier undersöker samband mellan variabler och har i forskningssammanhang, enligt Eriksson och Wiedersheim-Paul (1997), använts till följande:

- ?? Som illustration.
- ?? Som hjälpmedel att skapa hypoteser.
- ?? Som hjälpmedel vid aktionsforskning eller förändringsarbete.
- ?? Som hjälpmedel för att skapa en ny teori.

Gemensamt för fallstudier, oavsett syfte, är att de åtminstone har följande tre drag:

- ?? Betoning av aktörsrollen.
- ?? Studier av historiska förlopp.
- ?? Förmåga att kommunicera verkligheten.

Det finns några saker som forskaren skall ha i åtanke vid genomförande av fallstudier. Om två eller fler fall skall undersökas, skall jämförelser mellan fallen göras. Ett fall undersökt vid flera tidpunkter betraktas som separata fall och jämförelser skall göras. Det kan finnas liknande undersökningar som har gjorts tidigare, vilka går att relatera till i arbetet. Det är inte nödvändigt att relatera till andra fall, egna eller andras, utan forskaren kan istället jämföra med generella teorier och modeller. Slutligen går det att hävda att forskningen är av sådan karaktär att den är alltför speciell för att lämpa sig för jämförelser, det intressanta är istället förståelse för den specifika situationen (Eriksson, Wiedersheim-Paul 1997). Beträffande datainsamling är följande punkter viktiga att tänka på:

- ?? Vilka data är relevanta?
- ?? Vilka personer sitter på viktig information?
- ?? Hur skall data samlas in?
- ?? Hur skall insamlad data tolkas?

2.3.2 Kvalitet – Bedömningsmall

Eftersom vår uppsats syftar till att undersöka om en viss utvecklingsmetod fungerar som utvecklingsverktyg under vissa förhållanden, och att det här ställer vissa kvalitativa krav på vårt forskningsarbete, är det följaktligen nödvändigt att redovisa vilka kvalitativa krav vi ställer, och på vilket sätt vi uppfyller dem, på den prototyp som utvecklas. Vi har därför valt att se på problemet tvådelat, dels hur vi kan göra kraven på vår utvecklingsmetod mätbara, dels hur vi kan specificera kraven på den prototyp som kommer att utvecklas för att utvärdera metoden. Med hjälp av metodkraven går det att bygga upp en målbild för vad ett lyckat resultat är samtidigt som det går att kvalitetsbestämma den utvecklade produkten i sig. Därmed har vi en god grund för att se vilket arbete som utförts samt på vilket sätt det gjorts, vilket gör att vi har lättare att se vad vi gjorde för att nå målet i det specifika fall som uppsatsen behandlar.

I vår uppsats har vi valt att använda oss av certifieringskraven för SPI 2000 (Svensk Programvaru Industri) vilket (korrekt implementerat) kommer att borga för att produkten har en god kvalitet.

2.3.2.1 Tre typer av kvalitet i fråga om programvara:

Som teoretisk grund för SPI 2000 metodiken (Håkansson, 2000) har avstamp tagits från McCalls kvalitetsmodell som omfattar tre olika typer av kvalitet:

- ?? Extern kvalitet, som den uppfattas och beskrivs av användaren.
- ?? Intern kvalitet, som den uppfattas och beskrivs av utvecklaren.
- ?? Kvalitet (för kontroll i form av kvantitativ skala och metod som kan användas för mätning enligt ISO/IEC 14598-1).

Ovanstående definitioner på intern och extern kvalitet ligger till grund för en trestegstrappa som utgör den teoretiska grunden för hur det går att bedöma en programvaras kvalitet (se Figur 7 och Figur 8).

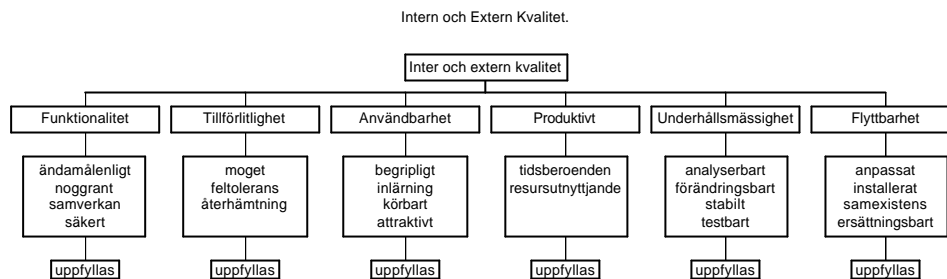
1. Sex plus en egenskaper, dessa kan i sin tur delas in i underkategorier:
2. Underegenskaper 31 st., vilka kan värderas och mätas med hjälp av:
3. Metrik, en kvantitativ skala och metod som kan användas för mätning av underegenskaper.

2.3.2.2 De 6 huvudegenskaperna:

SPI 2000 grundar sig på en vidareutveckling av McCalls kvalitetsmodell (Håkansson, 2000) vilken har inneburit en specifikation av sex huvudegenskaper hos en mjukvara som är lämplig att beaktas för att kunna designa för kvalitet.

- ?? **Funktionalitet**, en mängd av attribut som har sin grund i en uppsättning funktioner och deras specificerade egenskaper. Funktionerna är de som tillgodoser uttalade eller underförstådda behov.
- ?? **Tillförlitlighet**, en mängd av attribut som gör att programvaran kan upprätthålla sin prestandanivå under givna villkor för en given tidsperiod.
- ?? **Användbarhet**, en mängd attribut som avgör den prestation som är nödvändig för användning, och på den enskilda utvärderingen av en sådan användning, av en given eller underförstådd grupp användare.

- ?? **Underhållsmässighet**, en mängd attribut som avgör vilka resurser som är nödvändiga för att utföra specificerade ändringar.
- ?? **Produktivitet** (effektivitet), en mängd attribut som beror på förhållandet mellan det som programvaran utför och mängden av resurser som under givna villkor använts.
- ?? **Flyttbarhet**, en mängd av attribut som avgör vilka resurser som krävs för att flytta programvaran från en miljö till en annan.



Figur 7. Intern och Extern kvalitet med sex huvudegenskaper och 27 underegenskaper enligt ISO 9126-1 (Håkansson, 2000.).

2.3.2.3 Ytterligare en huvudegenskap.

På senare tid har en sjunde huvudegenskap lagts till:

Användarkvalitet, vilken speglar de egenskaper som påverkar användarens uppfattning av mjukvaran samt hur det är möjligt att använda den på ett effektivt sätt.



Figur 8 Kvalitet vid användning innefattar en huvudegenskap och fyra underegenskaper enligt ISO 9126-1 (Håkansson, 2000).

För att utvärdera de olika egenskaperna krävs att en utvärderingsnivå sätts för var och en. Först väljs en övergripande utvärderingsnivå för programvaran i samråd med beställaren enligt SPIs riktlinjer. När det här är fastslaget går det att bestämma vilken utvärderingsmetodik som bör utnyttjas för att bedöma de olika produktattributen (se bilaga 3).

2.3.2.4 Fem steg i utvärderingsmodellen:

Med ovanstående begreppsmässiga grund har SPI fastställt fem steg i sin utvärderingsmodell. Stegen skall utföras sekventiellt eftersom de bygger på varandra och varje steg måste slutföras innan nästa kan påbörjas.

Det första steget handlar om att fastställa utvärderingskraven. Det innebär att uppdragsgivaren och utvecklare kommer överens om de utvärderingskrav som skall gälla för det fortsatta arbetet.

Det andra steget innebär att de värderingar som skall gälla för ovannämnda värderingskrav specificeras. För att planera sin insats i förhållande till vad som beskrivs som en "tillräcklig nivå" rekommenderas att eventuella problem med systemet relateras till vad de får för återverkningar på omvärlden.

Som tredje steg i metoden utformas värderingarna i form av en utvärderingsplan baserad på specifikationen. Under denna aktivitet utformas, tillsammans med kunden, en specifikation som detaljerat beskriver de olika attribut som skall testas och vilka kravnivåer som skall gälla för respektive testpunkt. Samtliga attribut skall ha en så nära relation till de ursprungliga kraven som möjlig, för att öka förståelsen för varför just dessa krav testas.

Det fjärde steget är genomförande av utvärderingsplanen. Själva testningsarbetet utförs under denna fas och samtliga produktattribut skall värderas utifrån de riktlinjer som fastslagits under steg ett till tre. Testgruppen skall bestå av minst tre personer och det skall klart framgå vilka kriterier som är styrande. Det skall finnas en klart fastslagen metodik för testningen som delges testgruppen.

Som sista steg i metoden skall en sammanfattning av utvärderingen ske. Den innefattar även leverans av utvärderingsrapporten. För att en programvara skall uppfylla kraven för en SPI 2000 certifiering måste minst 22 *produktattribut* definieras och dokumenteras i en testspecifikation, varav minst tio under egenskapen *funktionalitet*. För att testspecifikationen skall godkännas måste samtliga underegenskaper för *funktionalitet* uppfylla kraven. Däremot räcker det att ett produktattribut godkänns för de övriga sex egenskaperna. Antalet godkända och verifierade produktattribut får inte understiga tolv stycken för de övriga sex egenskaperna.

2.3.3 Validitet och reliabilitet

Validitet och reliabilitet används när teoretiska föreställningar överförs till empiriska observationer. Validitet kan delas in i två olika aspekter: inre och yttre. Att mäta inre validitet är att säkerställa att rätt saker mäts. Den yttre validiteten syftar till att avgöra om det görs på rätt sätt. Förutom validitet finns det andra viktiga krav på ett mätinstrument. Ett sådant är reliabilitet, vilket är ett mått på om ett mätinstrument ger tillförlitliga och stabila utslag. För att uppnå god reliabilitet så skall en undersökning ge samma resultat oavsett vem som utför den. (Eriksson, Wiedersheim-Paul, 1997.)

2.3.4 Vår undersökningsstrategi

Vi har genomfört en fallstudie på ERV. Den här fallstudien syftade till att ligga till grund för att besvara vår frågeställning. Fallstudien bestod i huvudsak av observationer, intervjuer och kvalitetstestning av resultatet från utvecklingsarbetet. De olika aktiviteterna var till för att samla in primärdata för att bekräfta eller motsäga frågeställningen. För att bekräfta frågeställningen skall insamlingen av primärdata visa på att det skapats ett DSS av ett antal existerande system, som svarar mot de uppsatta kraven. Det var även en del av utredningen att visa utvecklingsmetodens relevans under skapandet, d.v.s. hur utvecklingsmetoden bidrog till skapandet av ett DSS. Datainsamlingen i fallstudien utfördes endast under utvecklingen av ett system, vilket utesluter jämförelser av olika fall.

Första aktiviteten i fallstudien var deltagande observationer, som sträckte sig över hela utvecklingsarbetet från planering till beslut tas att systemet är färdigt. Insamlandet av data skedde i dagboksform av utvecklarna. Den data som var relevant att samla in från observationerna var den som belyste utvecklingsmetodens styrkor och svagheter. Det finns olika perspektiv som metodens starka och svaga sidor kan belysas ur. Dels hur bra metoden är att arbeta utefter, dels hur lämplig den är för att utveckla ett DSS. För att påvisa hur lämplig metoden är för att utveckla ett DSS har vi sett till vad som gjordes för att uppfylla framgångsfaktorerna. För varje fas i utvecklingsmetoden (planering, förstudie etc.) summerade vi alla positiva och negativa upplevelser som speglar metodens för- och nackdelar. Summeringen gjordes av oss som forskare genom de deltagande observationerna där vi agerade utvecklare.

Reliabiliteten i den här aktiviteten var beroende av hur mycket utvecklarna visste om vad som skulle mätas i deras utvecklingsprojekt. Då vi som utvecklare var fullt medvetna om vad som var syftet med dagboken fanns risken att vi vinklade dokumenteringen av arbetet utifrån de svar som vi som forskare ville ha. Motsatsen till det här är att någon som inte vet någonting om vad forskningen handlar om skulle utveckla applikationen. Det här medför att utvecklaren inte är medveten om vad som är viktigt att dokumentera. Båda de här fallen sänker reliabiliteten i den insamlade data. I vårt fall var det viktigt att ge en så objektiv redovisning som möjligt för att öka reliabiliteten. Vi anser att vi kunde öka reliabiliteten då vi var medvetna om förutsättningarna.

För att ta reda på om den använda utvecklingsmetoden hade påverkat skapandet av ett DSS i någon riktning vände vi oss till beställaren av systemet för att få deras åsikter. Det här kunde således inte genomföras innan utvecklingsarbetet var avslutat. Den data som vi sökte skulle peka på om utvecklingsmetoden hade underlättat eller försvårat utvecklingsarbetet, och därigenom uppfyllandet av de uppsatta kraven.

För att besvara det här genomförde vi intervjuer med användare och beställare. Intervjuerna fokuserade på utvecklingsmetodens kvalitet och hur den hade påverkat kommunikation och samarbete mellan utvecklare och beställare, samt utvecklare och användare. Vi identifierade att kommunikation, inflytande, samarbete och kvalitet var viktiga huvudkategorier att utgå ifrån vid skapandet av intervjuguiden (se bilaga 2).

Då vi insåg att vi hade begränsade erfarenheter av intervjuteknik såg vi det som det mest lämpligt att arbeta utifrån en strukturerad intervjuguide. Vi försökte i största

möjliga mån att formulera frågorna så öppna som möjligt för att få fram perspektiv som var svåra att förutse. Eventuella följdfrågor under intervjun ställdes endast för att utveckla svar och inte för att belysa nya infallsvinklar. Personerna som vi intervjuade var de som hade haft stor inblandning i projektet och nära kontakt med utvecklarna. På grund av projektets omfattning identifierade vi två personer som var intressanta att intervjua; beställaren och den användare som var ansvarig för driften av MOBITEX-nätet. De intervjuades var för sig för att motverka att de skulle påverka varandra. Enligt Easterby-Smith (1991) krävs det dessutom än mer av intervjuledaren vid intervju av grupp. Vi beräknade att intervjuerna skulle ta ungefär en timme var i anspråk. En av oss fungerade som intervjuledare, under de både intervjuerna, medan de två andra antecknade.

Då användare och beställare inte i detalj visste vad utvecklingsmetoden omfattar, kunde de inte konkret säga hur den påverkat uppfyllandet av kraven. Den information de bidrog med kunde tillsammans med det vi visste tolkas för att besvara det här. Att vi behövde tolka den här informationen och länka den med utvecklingsmetoden bidrog till minskad reliabilitet. Ytterligare en faktor som påverkar reliabiliteten är hur skicklig intervjuaren är. För att öka reliabiliteten strukturerade vi intervjuerna i mesta möjliga grad.

För att bedöma den objektiva kvaliteten i utvecklingsarbetet utformade vi och använde en bedömningsmall enligt de riktlinjer som SPI har satt upp (se kap. 2.3.2). De tidigare metoderna för datainsamling är två subjektiva infallsvinklar, dels beställarens syn, dels utvecklarens syn.

Utgångspunkten för utformningen av verifieringskraven i kvalitetsbedömningsmallen, var de krav som beställaren specificerade i kravspecifikationen (se bilaga 4). För att synsätten för både den interna och externa kvaliteten skulle överensstämma så diskuterade beställare och utvecklare utkastet som utformats utifrån kravspecifikationen.

Kvalitetsmallen bygger på de sex plus en huvudegenskaperna och deras respektive underregenskaper (se bilaga 5). Utifrån huvudegenskaperna matchades motsvarande funktionalitet i kravspecifikationen, vilket ledde till ett antal mätbara attribut. I vissa fall gick det inte att formulera mätbara attribut, istället formulerades mer övergripande villkor som skulle uppfyllas, för att resultatet skulle bedömas som framgångsrikt.

För att förbättra reliabiliteten, dels för att uppnå objektivitet, dels för att uppnå de förväntade resultaten, var vi tvungna att sätta upp ramar för hur testverksamheten skulle bedrivas. Ramarna beskriver de yttre förutsättningar som skulle råda under testverksamheten, ifråga om teknisk plattform samt vilka grundläggande inställningar som skulle gälla för systemet. För att ytterligare förbättra reliabiliteten, under testerna, fanns det specificerade krav på vem som skulle utföra testerna och vem som skulle övervaka dem. Vi valde att följa SPIs rekommendationer (Håkansson 2000) beträffande val av testledare och testare. En av utvecklarna utseddes till testledare under testerna som genomfördes av tre personer, vilka var två utvecklare och en användare som ERV utsåg. Det kan tyckas konstigt att vi själva ingick i testpanelen men då SPI 2000 är en metod för *egendeklaration* av programvaror är det här en av grundtankarna. Utvecklarna skall själva kunna testa programvaran då det inte alltid finns en identifierad användare.

För att testerna skulle godkännas måste samtliga produktattribut för huvudegenskapen *funktionalitet* vara uppfyllda samt minst ett produktattribut från var och en av de övriga egenskapsgrupperna. SPI 2000 kräver att en genomförd testning dokumenteras i enlighet med standard. Vi har istället valt att redovisa resultaten i resultatdelen av vår uppsats.

Ovanstående beskriver hur vi gick tillväga för att samla in primärdata. Primärdata består av interna aspekter, såväl som externa och objektiva. Vi har således försökt att täcka in så många aspekter som vi anser att fallstudien tillåter. Rätt utformade täcker de här metoderna in den totala mängden information, som finns för att besvara den uppsatta frågeställningen. Validiteten kunde ha förbättrats om det hade funnits möjlighet att använda utvecklingsmetoden i flera projekt och sedan jämfört dem.

Vi anser inte att vår undersöknings validitet minskat genom att vi avgränsat oss till att genomföra de fem första stegen i vår utvecklingsmetod. Visserligen förblir de sista två stegen otestade men vi anser inte att det här påverkat vår möjlighet att erhålla ett relevant resultat.

3 Resultat

Vi har utfört deltagande observationer, intervjuer och kvalitetssäkring enligt SPI. De deltagande observationerna grundar sig i de dagböcker som vi som utvecklare skrivit under utvecklingen av prototypen. Resultatet från observationerna fokuserar på utvecklarnas syn på hur utvecklingsmetoden har främjat framtagandet av applikationen. Vidare fokuserar resultatet från intervjuerna på beställaren och användarnas syn på utvecklingsmetoden. Slutligen skall resultatet från kvalitetssäkringen visa på om systemet uppfyller testspecifikationen, vilket innebär att resultatet från utvecklingsmetoden genomgår en kritisk granskning.

3.1 Utvecklarnas sammanställda dagböcker

Dagboken är en sammanställning av den dokumentation som vi som utvecklare har utfört under arbetets gång. Om inget annat framgår skall ”vi”, i dagboken, tolkas som att det är utvecklarnas åsikter som förs fram.

3.1.1 Planeringsfas

När vi påbörjade utvecklingsarbetet tillsammans med ERV hade de sammanställt en projektbeskrivning som beskrev en del riktlinjer för vad uppdraget bestod i. ERV hade konstaterat att det fanns ett problem samt att de hade skapat sig en uppfattning om vad lösningen på problemet kunde vara. I projektbeskrivningen hade ERV fokuserat på en typ av lösning, vilket var den ERV upplevde att de ville ha. I den inledande diskussionen med ERV var vår intention som utvecklare i planeringsfasen att försöka vidga ERVs uppfattning till en mer allmän bild av problemområdet. Målet med den här diskussionen var att kunna ta reda på vad ERV verkligen ville ha och sammanfatta det här i ett övergripande mål. Vår gemensamma bild av målet var att skapa en applikation som på ett enklare och smidigare sätt kunde hjälpa till att positionera mobilerna för att finna fel i driften av ett MOBITEX-nät. Vid den här tidpunkten bestod det här arbetet i att sammanställa data från ett antal existerande system, vilket ERV upplevde som tidsödande samt att det var svårt att skapa sig en överblick med hjälp av de tillgängliga data som erhöles.

Under planeringsfasen diskuterade vi även med ERV hur vi hade tänkt lägga upp projektet, dels aktivitetsmässigt, dels tidsmässigt. Resultatet av den här diskussionen var en tidsplan (se bilaga 6) där tidsramar sattes för de olika faserna. Vi tyckte att planeringsfasen förlöpte smidigt, men vi upplevde att ERV var väl specifik i sina beskrivningar och relaterade mycket till tekniken. I det här skedet utgjorde utvecklingsmetoden ett stöd då den styrde in oss till att se saker i ett vidare perspektiv.

3.1.2 Förstudie

I förstudien utgick vi ifrån det övergripande målet för att försöka skapa oss en bättre bild av hur problemområdet såg ut. För att skapa en bättre bild samlade vi in väsentlig information om beslutsprocesserna genom att ha ett gruppmöte med ERV. Innan vi kunde genomföra gruppmötet var vi först tvungna att identifiera rollerna inom problemområdet, vem som var beställare respektive användare. Organisationens struktur kring problemet identifierade ett antal användare och en beställare. Något vi även var tvungna att göra innan mötet var att reda ut de nya begrepp från användaren och beställarens verklighet som framkommit. Då deras värld innehåller specifik teknik som vi inte stött på tidigare tog det tid att förstå begreppen och sambandet mellan dem.

Vi valde att ha gruppmötet med tre användare dels för att de tillsammans skulle komma överens om en gemensam uppfattning av problemet ur ett användarperspektiv, dels för att undvika att beställaren skulle blanda in för tillfället onödiga politiska faktorer. Genom gruppmötet kartlade vi hur de arbetar idag och därigenom kunde vi avgöra vilka av de fyra faserna i beslutsprocessen som systemet skulle stödja. Användarna behövde i första hand stöd under fasen för informationsinsamling för att kunna lösa sina problem.

Fasen för informationsinsamling innebär att användaren försöker konstatera huruvida de organisatoriska målen uppfylls eller inte. För ERVs del innebär det här att ta reda på om MOBITEX-nätet fungerar felritt eller inte. När problem uppstår i nätet försöker användarna att identifiera symptomen och deras omfattning. För att utreda det här söker användarna information som kommer från ett antal textbaserade system. Efter att det här är utfört kan användaren avgöra om det verkligen existerar ett problem och vilken omfattning det har. Vi tog även reda på hur väl det går att strukturera den information som ligger till grund för att lösa problemen med MOBITEX-nätet. De sätt som användaren samlar in information på är strukturerat och går att dela in i två delproblem, mobiler och basstationer. Den som är ansvarig för att finna lösningen för problemet är den användare som är ansvarig för nätet vid tillfället då problemet uppstod. Användarna som sköter driften av nätet är mycket kunniga inom området och har stor kunskap om dels den bakomliggande tekniken, dels systemets uppbyggnad. Den kunskap som användarna besitter är avgörande för att de skall kunna fatta beslut med hjälp av den information som finns att tillgå. Ett mål för användarna var att det inte skulle behövas lika hög kompetens för att kunna söka information i framtiden.

Vi upplevde att det var svårt att kartlägga beslutsprocessen under gruppmötet. Det här beror troligtvis på att vi hade problem med att få användarna att ena sig under mötet. Under gruppmötet visade ett flertal av användarna tendenser till att fokusera på specifika lösningar, ett sådant exempel var datalagringen. I ett försök att få användarna att bortse från tekniska lösningar, vid den här tidpunkten, påtalade vi att det var för tidigt att fastslå hur data skulle lagras, vilket accepterades.

För att vidare utforska den rådande situationen och få uppslag till lösningar övergick vi till att studera de resurser för informationsinsamling som fanns tillgängliga. För att ta reda på hur resurserna gick att utnyttja studerade vi befintliga system och dokumentation om dem. Ett förslag togs fram över hur vi uppfattade den nuvarande situationen och hur den förbättrade situationen skulle se ut, vilket demonstrerades genom en rik bild (se bilaga 7). I den rika bilden lade vi fokus på förändring då den beskriver en övergripande omvandling mellan två situationer i en organisation. Den ena sidan av bilden beskriver situationen idag medan den andra beskriver hur vi uppfattar att situationen kan bli bättre för användarna. De punkter vi tog fasta på under förstudien, som även framgår i den rika bilden, var att samla data från flera existerande system under ett gemensamt skal, att presentation av data även skall vara grafisk vilket underlättar tolkningen och att det skall gå att lagra data, vilket är funktionalitet som inte fanns.

3.1.3 Definition av system

I planeringsfasen såg vi till de symptom som användarna kände, vidare i förstudien utredde vi orsakerna till symptomen för att slutligen komma fram till ett första förslag på vad som skall göras för att råda bot på dem. Den rika bild som förstudien resulterade i, ligger till grund för utformningen av en eller flera systemdefinitioner. Utifrån vår rika bild konstruerades två förslag till systemdefinition. Vi valde att använda oss av VATOFF för att säkerställa systemdefinitionernas validitet. Då användarna är vana vid beskrivningar av teknisk karaktär gjorde vi bedömningen att den här metoden för validering skulle lämpa sig bäst. Genom att använda VATOFF ökades användarens förståelse och engagemang för utvecklingsarbetet och tid kunde sparas.

Utöver det som redan framkommit beskriver systemdefinitionen tekniken som skall användas och den omgivning som systemet skall verka i. Reaktionerna från beställare och användare var att systemdefinitionen är väsentlig då de väl känner till begreppet och den fyller en viktig funktion som ett bevis på att utvecklare och beställare är i samförstånd. VATOFF-dokumentet ställde beställaren sig mer tveksamt till då de inte uppfattat syftet med det. Det här framkom då användare och beställare menade att systemdefinition och VATOFF i stort sett beskrev samma sak.

Beställarens val av systemdefinition innebar att det framkom nya fakta som påverkade utformningen. Det här innebar att vi tvingades omarbeta de systemdefinitioner vi tidigare gjort. Vi upplevde att det var givande att ha presentera två systemdefinitioner trots att de var relativt lika, då det här resulterade i att beställaren tvingades jämföra de två och därigenom tillförde aspekter som inte tidigare framkommit. När vi omarbetat definitionerna valde beställaren en av systemdefinitionerna.

Utifrån systemdefinitionen kunde vi formulera ett antal funktionella krav som skulle ingå i kravspecifikationen. Då kravspecifikationen fungerar som ett kontrakt mellan beställare och utvecklare var det viktigt att beskriva konkreta krav utifrån vad som är möjligt att genomföra tekniskt. En del av de tekniska kraven är redan givna genom det tidigare arbetet i den här fasen men det som ytterligare måste klargöras är vad utvecklingsverktygen och den tekniska plattformen tillåter samt hur kommunikationen skall ske med de existerande systemen. Det är först under den här fasen det är aktuellt att specificera tekniska krav. Användarna var intresserade att göra det här tidigare men vi upplevde att det var relevant att diskutera de tekniska kraven först i samband övriga kraven.

Efter att kravspecifikationen sammanställts hade vi en klar bild av vad som skulle utföras. Vi upplevde att vi hade tillräckligt med information för att kunna gå vidare till nästa fas. Den här uppfattningen förstärktes ytterligare av att beställare och användare spontant gav uttryck för att den stämde väl överens med deras tankesätt.

Utifrån kravspecifikation utformade vi, tillsammans med användare och beställare, en testspecifikation som skulle ligga till grund för verifieringen av systemet. Vi använde SPIs metod för systemverifiering, SPI 2000 (se kap. 2.3.2) för att göra kraven mätbara. Tillsammans med beställaren och en användare identifierade vi att utformningen av testspecifikationen skulle följa *Nivå D* (se bilaga 3). Uppförandet av en testspecifikation uppfattades som positivt av användarna och beställaren speciellt då de själva medverkade i framtagandet.

3.1.4 Konceptualisering

Eftersom vi skulle utveckla ett beslutsstödssystem var den övergripande arkitekturen i princip redan given: modell, data och gränssnitt. Vårt arbete bestod därför i att fördela kraven på systemet på den givna arkitekturen, samt att specificera hur arkitekturen skulle realiserats.

Efter att ha genomfört de tidigare faserna i utvecklingsmetoden tyckte vi att vi hade goda förutsättningar för att snabbt komma igång med prototypingen. Vi kunde även se att det nära samarbetet med användarna hade gett resultat. Utvecklarna kunde lägga mer energi på att lyckas bra i första försöket istället för att på ett mer ostrukturerat sätt försöka nå samma resultat. Samtidigt var det lätt för användarna och beställaren att förstå vad utvecklarna åstadkom och hur arbetet gjorde framsteg. Vårt nära samarbete med användarna bidrog till att det framstod som självklart att prototypingen skulle börja med användargränssnittet.

För att ta fram layouten av de olika användargränssnitten använde vi oss av pappersprototyper. Det går ut på att gränssnittet designas med utklippta delar av papper. Användaren får sedan prova gränssnittet och ge respons som resulterar i vissa förbättringar. När användaren känner sig nöjd kan kodningen av gränssnitten börja. Ett gränssnitt kan vara internt och externt konsekvent (Joiner 1998). Intern konsekvens innebär att gränssnittet uppvisar ett likartat utseende och beteende inom programmet. Extern konsekvens innebär att programmet påminner om ett annat program till utseende och beteende. ERV ville att programmet skulle vara internt konsekvent samt att det skulle vara externt konsekvent med redan existerande testverktyg. För att uppnå intern och extern konsekvens i gränssnitten var det viktigt att på ett mycket detaljerat plan specificera informationen om dem.

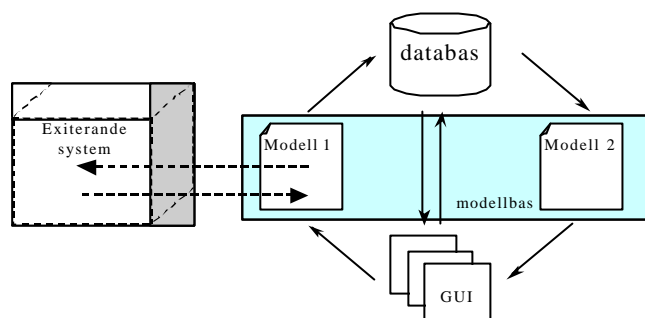
För att plocka fram den information som användaren upplever som självklar men svår att beskriva, kändes det nödvändigt att visualisera det tänkta användargränssnittet. Fördelarna vi upplevde med att utforma prototyperna för användargränssnittet på papper, var att det gick snabbt att genomföra förändringar samtidigt som de var tillräckligt realistiska för att användaren skulle få god insikt i hur det tänkta användargränssnittet skulle se ut och fungera. Vi tror även att den här processen har bidragit till att öka användarnas känsla för delaktighet och engagemang i utvecklingsprocessen.

Utvecklingen av användargränssnittet i det valda verktyget (Tcl/Tk) följde den framtagna pappersförslagan, vilket medförde att vi sparade tid då den grundläggande layouten i användargränssnittet inte behövde korrigeras. Kunden kom även i det här läget med ändringsförslag som tidigare hade varit svåra att förutse. Förändringarna var inte något problem att genomföra då de var av mindre omfattning. Då pappersprototyperna på ett effektivt sätt tagit fram den övergripande layouten var det småförändringar som återstod. Vi upplevde att det här främjade användarens vilja att påpeka nödvändiga förändringar vilket i sin tur bidrog till att öka användarnas delaktighet. Som utvecklare kände vi att deras engagemang medförde att vi i vår tur satte värde på förslagen till förändring.

När kunden godkänt användargränssnittet fanns nu en första användbar version av prototypen. För att kunna gå vidare med att utveckla funktionaliteten var det viktigt att identifiera vilka beståndsdelar som skall ingå i modellbasen (se kap. 1.2.1). Vi

identifierade två modeller, dels en som skall samla in data, dels en som skall presentera data på ett så överskådligt sätt som möjligt. Vi upplevde att det fanns tillräckligt med information i kravspecifikationen för att kunna göra det här på ett okomplicerat sätt. Användarna kände igen modellen utifrån det som tidigare diskuterats, vilket vi tror bidrog till att öka deras känsla av ägandeskap.

I den fortsatta prototypingen kände vi som utvecklare att arkitekturen fungerade som mall att arbeta efter och underlättade även arbetsfördelningen. Framtagandet av modellbasen baserades till stor del på de användarscenarion som framkommit under designen av gränssnitten, vilka gav en fingervisning om att det skulle finnas två modeller, en för insamling av data till systemet och en för bearbetning av data.



Figur 9 Arkitektur över prototypen.

Som Figur 9 visar arbetar modell 1 enbart mot existerande system för att samla in den relevanta data vilken sedan skall skickas för lagring i databasen. Modell 2 i sin tur bearbetar insamlad data bl.a. för presentation i det grafiska gränssnittet. Då modellbasen är mycket central i ett DSS har vi lagt mycket arbete på att få den att uppfylla kraven. En fördel som vi upplevde var att det var fullt möjligt att arbeta med utvecklingen av båda modellerna samtidigt.

Vi anser att evolutionär prototyping har bidragit till att bibehålla enkelheten i systemet. Då systemet växer fram stegvis undviks att onödig funktionalitet inkluderas, fokus hamnar istället på den centrala funktionaliteten. De svårigheter som vi har upplevt är att vi saknade ett lämpligt sätt att dokumentera de förändringar som framkommit under prototypingen, särskilt vid några tillfällen då det framkom många förslag till förändringar under en kort tidsrymd. Det här resulterade i att alla inblandade parter kände sig osäkra på vilka beslut som hade tagits kring förändringarna. Problematiken med prototyping i det här sammanhanget kan vara att det är svårt att göra många förändringar samtidigt då det kan få konsekvenser i andra delar av prototypen som inte är möjliga att förutse. Sammantaget ställde det här stora krav på oss som utvecklare att vi skulle vara extra strukturerade och effektiva.

Samtidigt som modellbasen växte fram var vi tvungna att planera för hur data skulle struktureras i databasen. Vilken data som skulle sparas var redan fastställt i de tidigare faserna men inte hur den skulle vara organiserad. När det blev aktuellt att strukturera databasen upplevde vi att den diskussion som uppstod fördröjde arbetet. Det kändes som att organiseringen av databasen skulle skett tidigare i konceptualiseringsfasen. På grund av det här fördröjdes arbetet och tid gick åt för att skapa en gemensam bild av strukturen då de olika utvecklarna skapat sig egna idéer under tiden.

Iterationsprocessen har ämnat till att utöka prototypen i linje med kravspecifikationen samt att tillgodose de nya behov som framkommit under utvecklingens gång. I och med iterationsprocessen har det framkommit nya behov från användarna som har tillgodosetts. Som utvecklare har det känts naturligt att tillgodose de nya behov som framkommit. Eftersom ett nytt behov oftast uppkommit som reaktion på det utvecklaren konceptualiserat har det varit relativt enkelt att tillgodose de nya behoven. Det här på grund av att problematiken just då har varit aktuell för utvecklaren. Men det har även framkommit behov som vi tillsammans med användare och beställare bedömt att vi inte kunnat tillgodose då vi har varit måna om att hålla de uppsatta tidsramarna. Vi upplevde att de flesta förändringar gällande användarens behov har gällt presentationen av data. Dessa behov gick inte förutse då det var extremt svårt att på förhand specificera hur data borde presenteras för att ge ett bra beslutsunderlag. Det är värt att poängtera att det noggranna förarbete som utförts inte är att förkasta då det här inneburit att vi som utvecklare kunnat koncentrera oss mer på de behov som inte gick att utreda på förhand.

3.1.5 Utvärdering

Under utvärderingen skall systemet testas utefter testspecifikationen för att säkerställa att det uppfyller alla krav. Som utvecklare använde vi oss av testspecifikationen redan under konceptualiseringsfasen, dels för att kunna bedöma vilket arbete som skulle utföras, dels hur lång tid det återstående arbetet fick ta. Den formella utvärderingen präglades av en mer sammanhållen och strukturerad testverksamhet där både vi som utvecklare och användarna ingick i testgruppen.

Själva testverksamheten inleddes med att representanter för utvecklarna utförde tester enligt testspecifikationen, därefter fick användare och beställare genomföra testerna under ledning av utvecklarna.

Som utvecklare upplevde vi att vi var förberedda på vad som skulle göras och på så vis påverkade vi inte resultatet av testerna negativt genom att försvara vårt arbete. Vi upplevde också att användarna uppskattade att vara en del av utvärderingsprocessen. De kände delaktighet i utvärderingen men uttryckte även tacksamhet för att ansvaret för hela testproceduren inte låg hos dem.

3.1.6 Kommentarer till dagbok

Då de två sista faserna av vår utvecklingsmetod inte innefattas i vår fallstudie har vi följaktligen inga resultat att presentera. Vi kommer nu som utvecklare ge våra synpunkter på hur metoden upplevts stödja oss i vårt arbete, samt vilka brister och tveksamheter vi upplevt.

För det första upplever vi att vår prototyp av systemet kräver mindre av användarna såtillvida att informationen blir mer lättillgänglig. Det här åstadkoms dels genom att endast relevant data presenteras, dels genom att användarna inte behöver besitta lika mycket kunskaper om tekniken. Vi tror även att användaren på det här sättet kan spara tid i sitt informationssökande.

För det andra anser vi att kombinationen av SSM och prototyping har främjat framtagandet av systemet. Att använda SSM för att utföra planering, förstudie och systemdefinition innebar att vi fick en god insikt i användarnas arbetssätt, vilka

problem de upplevde och möjliga lösningar till problemen. Vi som utvecklare upplevde även på ett tidigt stadium att vi etablerade en god relation med användare och beställare. När vi övergick till prototypingen i konceptualiseringsfasen hjälpte denna metod användare och utvecklare att beskriva fenomen som är svåra att uttrycka med ord och därmed inte kommit fram i de tidigare faserna.

För det tredje anser vi att vi har gjort rätt saker på rätt sätt. Då vi anser att prototypen uppfyller kravspecifikationen visar det på att vi har gjort saker på rätt sätt. Användarna har endast visat positiva reaktioner på systemet, vilket visar på att vi har utfört rätt saker, alltså det som användarna förväntat sig.

3.2 Utvärdering utvecklingsarbete enligt uppdragsgivarna

Då användarna och beställaren är involverade i utvecklingsprocessen är det viktigt att även ta del av deras uppfattningar om utvecklingsarbetet för att få en mer heltäckande utvärdering av metoden. För att ta reda på hur utvecklingsmetoden uppfattas av beställaren och användarna har vi valt att genomföra intervjuer med dem. Vi ville ta reda på vad de anser om metodens kvalitet, och hur den påverkar kommunikation och samarbete, mellan dem och utvecklare.

Under kategorin kommunikation inriktade vi oss på att få ett underlag för utvärderingen av hur de intervjuade upplevt kontakten med utvecklarna, samt om graden av kontakt hade skilt sig åt under de olika utvecklingsfaserna. Frågorna skulle även undersöka om beställaren och användarna känt sig tillräckligt informerade under utvecklingsarbetet, och om utvecklarnas kunskaper om tekniken hade upplevts tillräckliga för att föra en fruktsam dialog.

Användaren tyckte att det var lätt att ta kontakt med oss som utvecklare under arbetets gång. Han såg det även som positivt att systemet utvecklades hos dem och att det alltid gick att få tag i utvecklarna. Han upplevde inte heller att vårt sätt att arbeta var något hinder för kommunikationen, men att det var något svårare i början då vi som utvecklare inte hade hunnit få så stor kunskap om verksamheten och den teknik som används (MOBITEX). Allt eftersom utvecklarna lärde sig om organisationen och dess miljö krävdes det mindre insatser för att kunna kommunicera på ett enkelt och smidigt sätt. Användaren tycker inte att han har fått för lite information om projektets organisering, men att det hade underlättat för honom om kommunikationen kring det här hade skett på skriftlig väg. Han trodde samtidigt att det är extra viktigt att ta det här i beaktande i ett större utvecklingsprojekt. Vidare kunde inte användaren identifiera några faser i projektet, men upplevde inte det här som något nödvändigt. Användaren tror inte att tekniken har haft någon större betydelse för kommunikationen då projektets omfattning inte krävt så djupa kunskaper om MOBITEX och de relevanta existerande systemen.

Beställaren tyckte liksom användaren att det över lag har varit enkelt att få kontakt med oss som utvecklare. Då kravspecifikationen innehöll den relevanta informationen men på en övergripande nivå upplevde han det som positivt att vi var på plats under utvecklingsarbetet. Det här bidrog till ökad kommunikation och han kände att det var lätt att få nödvändig överblick av projektet. Beställaren upplevde att det var i början av projektet som det krävdes mer kontakt med oss som utvecklare då det tar ett tag innan vi kunde förstå varandra fullt ut. Det är då naturligt att det krävs mer kommunikation i inledningen av ett projekt. Beställaren upplevde att dialogen med

oss som utvecklare fungerade på ett bra sätt, vilket märktes genom att vi lade ner mycket energi för att förstå situationen. Samtidigt så uppfattade beställaren att vårt sätt att arbeta framstod som väl strukturerat. Han tyckte inte heller att tekniken var något hinder för oss som utvecklare.

För att utreda de intervjuades upplevda inflytande formulerades frågor angående hur väl utvecklarna lyssnat till deras krav och önskemål. Hur beställaren och användaren uppfattar sitt inflytande i utvecklingen är nära sammankopplat med hur väl samarbetet fungerar. För att belysa samarbetet så valde vi att ställa det här i relation till om systemet istället skulle utvecklats av externa konsulter, respektive ERVs egna utvecklare.

Användarens uppfattning om inflytande är att vi som utvecklare verkligen har tagit åt oss av de idéer som han har lagt fram, vilket även märks i kravspecifikationen. Något som han har uppfattat som negativt är att det saknas formella tillvägagångssätt för hur dokumentationen av nya idéer och förändringar skall ske. Resultatet har blivit att vissa muntliga överenskommelser har fått upprepas samt att det ibland har varit svårt att kunna dra sig till minnes om det är något som han har tagit upp med oss som utvecklare eller om det bara har varit något som han har funderat över personligen. Användaren har upplevt att vi har arbetat självständigt, men att vi har sökt upp honom vid behov för att få nödvändig information.

Användaren tror att om uppdraget hade gått till en konsult hade vederbörande troligtvis försökt följa mer av ERVs standarder. Resultatet hade troligtvis blivit detsamma men att projektet hade producerat fler dokument enligt ERVs standard. Vidare så trodde användaren att resultatet av att utveckla projektet inom ERV hade blivit det samma rent samarbetsmässigt, men att han kunde se både för och nackdelar inflytandemässigt. Fördelen skulle kunna bli att det tillkommer funktionalitet som inte vi som utvecklare kan se nyttan av. Nackdelen är dels att den nödvändiga funktionaliteten drunknar i mängden, dels att projektet kanske aldrig skulle ta slut.

Beställaren anser att samarbetet har fungerat på ett bra sätt och att vi inte har missat något väsentligt. Han anser vidare att det var lätt att få fram det som han ansåg var viktigt med projektet. Beställaren tycker inte att det har funnits några barriärer mellan oss som utvecklare och honom.

Om ERV hade låtit en konsult utföra uppdraget istället trodde beställaren att det hade resulterat i att de hade fått mer dokumentation enligt vad som är standard på ERV. Vanligtvis brukar de ställa större krav på utvecklare samt att de får mer specifika riktlinjer för hur projektet skall bedrivas. Beställaren trodde att ERVs utvecklare skulle göra arbetet något snabbare då de är mer bekanta med omgivningen och vad som krävs. Liksom användaren så trodde beställaren att mer förslag på funktionalitet hade framkommit om de stått för utvecklingen själva.

För att kunna utreda kvaliteten konstruerades frågor där beställaren och användarens uppfattning om hur väl arbetet genomförts, och om det var rätt arbete som utförts. Vi ville även få fram en mer allmän bild av vad kvalitet innebär för de intervjuade och hur väl prototypen motsvarar den här bilden.

Användaren menar att den upplevda kvalitén visar sig i det som projektet producerar. Han ansåg vidare att det är viktigt att system byggs på rätt sätt samt att det finns en kontroll på att det är rätt sak som produceras. Användaren tyckte att vår utvecklingsmetod framstod som lagom strukturerad, men att den borde producera lite mer dokument i ett tidigare skede. Vidare ansåg han att metoden innehåller rätt delar samt att de kommer i en rimlig ordning. Han framhåller även att det är viktigt att ha tillgång till duktiga programmerare för att uppnå god kvalitet. Helhetsintrycket av metoden är att den framstår som pålitlig för det ändamål som den har använts till, men att den kanske skulle ha anpassats lite mer till hur han är van att arbeta.

Beställaren tycker bl.a. att kvalitet hänger ihop med hur väl kravspecifikationen speglar vad han vill ha ut ur ett projekt. Han tycker att vi som utvecklare har avsatt lagom med tid för att framställa kravspecifikationen. Vidare anser han att kvalitet i ett utvecklingsprojekt kan bedömas utefter hur bra produkten fungerar och hur enkelt det är att modifiera den. Då han inte har jobbat lika nära med oss som användaren har han lite svårt att göra ingående bedömningar i projektets kvalitet, men han anser att vi har framstått som noggranna och strukturerade i vårt arbete.

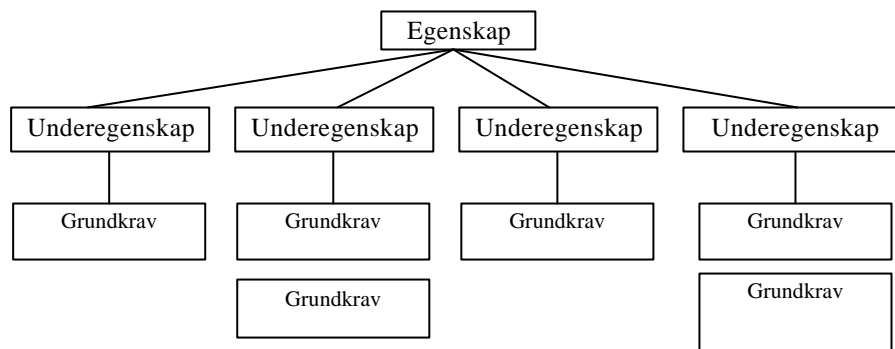
Som forskare är vi även intresserade av att ta reda på vad användarna hade för bild av systemet innan utvecklingsarbetet inleddes. Var det möjligt för dem att definiera hela systemet eller fanns det delar som var mer eller mindre svåra att specificera. Vi ansåg även att det var relevant att ta reda på om de tyckte att vi hade fokuserat på den teknik som skulle användas i utvecklingen.

Användaren ansåg sig ha en ganska så klar bild av vad systemet skulle komma att omfatta i form av funktionalitet och gränssnitt. Han hade på förhand önskemål som var ganska detaljerade t.ex. hur vissa gränssnitt skulle se ut och hur data skulle lagras. Användaren hade även från början förslag på vilka verktyg som skulle användas i utvecklingen. Vidare så trodde han att det skulle ha varit svårt att definiera hela systemet från början utan att använda sig av en inkrementell utvecklingsmetod.

Beställaren hade en ganska bra helhetsbild av systemet från början men hade till skillnad från användaren inga specificerade bilder av systemet. Han trodde även att det skulle vara svårt att göra en färdig design av systemet. Den största orsaken till det här ansåg han var att ERV inte skulle kunna specificera allt in i minsta detalj. Vidare var han inte säker på vilka begränsningar den valda tekniken skulle utgöra. Beställaren menar även att han inte upplevde det som att utvecklingsmetoden fokuserade på vilken teknik som skulle användas, inte heller att den satte några speciella begränsningar för vilken teknik som skulle kunna användas.

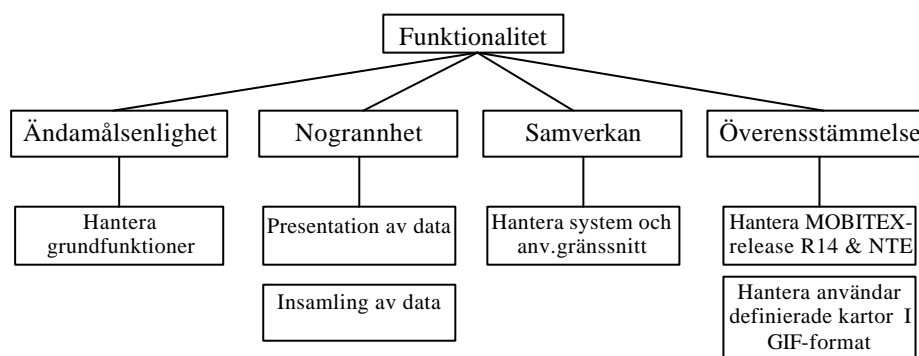
3.3 Utvärdering av kvalitet enligt kvalitetsmall

Presentationen av resultatet från kvalitetstestningen enligt SPI 2000 kommer att ske med utgångspunkt i de sex plus en egenskaperna: *funktionalitet*, *tillförlitlighet*, *användbarhet*, *produktivitet*, *underhållsmässighet*, *flyttbarhet* och *användarkvalitet*. Varje egenskap kommer att beskrivas utifrån dess produktattribut och grundkrav i den mån det inte finns några produktattribut.



Figur 10 Relationer mellan delarna i testspecifikationen.

3.3.1 Funktionalitet



Figur 11 Del av testspecifikationen: egenskapen funktionalitet.

För att kunna godkänna programvaran enligt SPIs krav på *funktionalitet* måste samtliga av produktattributen vara uppfyllda.

Testpanelen av systemet kom fram till att det går att starta en loggning. Att starta loggningen innebär att trycka på "start" -knappen (se bilaga 8) när samtliga förutsättningar är uppfyllda. Vidare går det att stoppa en loggning genom att trycka på "abort" -knappen, under förutsättning att en loggning pågår.

Förutsättningen för att kunna starta en loggning är att användaren väljer en tidsperiod för loggningen. Med tidsperiod avses ett datum och ett klockslag för startpunkt respektive slutpunkt. Det går även att välja att loggningen skall starta omedelbart och/eller att den skall hålla på tills användaren avbryter den. Testpanelen lyckades utföra den här operationen.

Ytterligare en förutsättning för att kunna starta en loggning är att användaren väljer mellan en till tio mobiler som skall loggas samt hur ofta det här skall ske. Mobilerna väljs genom att ID-nummer för varje mobil (MANnr) matas in i ett textfält.

Tidsintervallet (pollintervallet) ställs genom att användaren väljer ett värde i en lista. Testpanelen hade inga svårigheter att utföra det här.

När loggningen startas kommer användaren till ett gränssnitt som visar den pågående loggningen, vilket genomfördes av testpanelen. Där går det att se varje post som skapas under loggningen i realtid. Från det här gränssnittet går det att komma vidare till det tredje gränssnittet som presenterar loggningen på en karta. Genom att trycka på "map"-knappen lyckades testpanelen att ta sig till det här gränssnittet. Användaren kan även välja mellan olika delkartor att visa resultatet på, vilket genomfördes av testpanelen.

Genom att välja en tidpunkt i gränssnittet med kartan går det att visa samtliga mobilers position. Testpanelen upptäckte några fel i prototypen när flera mobiler loggades och det uppstod problem med kommunikationen till NCC. Genom att markera en specifik mobil i en lista och ställa in ett tidsintervall går det att visa det här för den valda mobilen på kartan. Samtidigt som en mobil väljs ur listan kommer resterande mobilers position och status att visas för den valda tidpunkten. Testpanelen gav godkänt på de här funktionerna.

Användaren skall även kunna zooma kartan för att studera ett specifikt område närmare. Zoomfunktionen aktiveras genom att en kryssruta markeras och sedan kan användaren dubbelklicka på något av de nu markerade områdena. Genom att testpanelen godkände även denna funktionalitet har de även gett systemet godkänt för samtliga de krav som underregenskapen *ändamålsenlighet* beskriver.

Nästa underregenskap, *noggrannhet*, sätter upp kraven för insamling och presentation av data. Testpanelen kom fram till att systemet kan visa en basstation på kartan med 1 km noggrannhet. De kom även fram till att den maximala avvikelsen från pollintervallet ligger utanför det angivna gränsvärdet, vilket medför att produktattributet underkändes. Samtliga attribut som skall lagras för en loggning, såsom mobilens nummer och basens nummer, lagrades inte på ett korrekt sätt och det här produktattributet fick således inte heller godkänt. Systemet fick även underkänt för nästa attribut då avvikelsen från valt tidsintervall var utanför tio minuter.

Samverkan är den underregenskap som bl.a. mäter systemets förmåga att hantera system och användargränssnitt. Testpanelen kunde inte kontrollera att systemet belastar MOBITEX-nätet med mindre än ett kommando per mobil och minut på grund av de tidigare felen.

Sista underegenskapen i *funktionalitet*, som kallas *överensstämmelse*, mäter systemets förmåga att hantera olika systemversioner av MOBITEX samt användardefinierade kartor. Båda de här grundkraven är begränsade av förutsättningarna för testet varför inte all funktionalitet kan testas. Testpanelen ger godkänt för de båda då de uppfyller kraven utifrån förutsättningarna för testet.

Grundkrav	Produktattribut	J/N
Hantera grundfunktioner	Logga mobil (starta/stoppa)	I
	Visa alla mobilers status	I
	Välja tidsperiod för loggning	I
	Välja antal mobiler för loggning	I
	Välja pollintervall	I
	Visa loggning för mobil	I
	Presentera loggning på karta	I
	Presentera alla mobilers position för specificerad tidpunkt	N
	Presentera en mobil för ett valt tidsintervall	I
	Visa alla mobilers status	I
	Zoomfunktion för karta	I
	Presentation av data	Basstation ska ritas ut med en noggrannhet av 1 km
Insamling av data	Maximal avvikelse från pollintervall får vara 5 min	N
	Maximal avvikelse från valt tidsintervall får vara 10 min	N
Lagring av data	Mannr, BASnr, mobilstatus, datum och tid ska lagras vid loggning	N
Hantera system och användargränssnitt	Maximalt belasta NCC med i genomsnitt ett kommando per minut och mobil	N
Hantera MOBITEX-release R14 & NTE	Hantera MOBITEX-release NTE	J

Figur 12 Resultat från testning av funktionalitet.

Vi som testledare tyckte att det här var ett värdefullt sätt att testa hur väl systemet uppfyller kraven. Vi fick en bra sammanställd bild av vad testpanelen anser om systemet då testningen utfördes. Samtidigt var det enkelt att genomföra de olika delmomenten som testet består av.

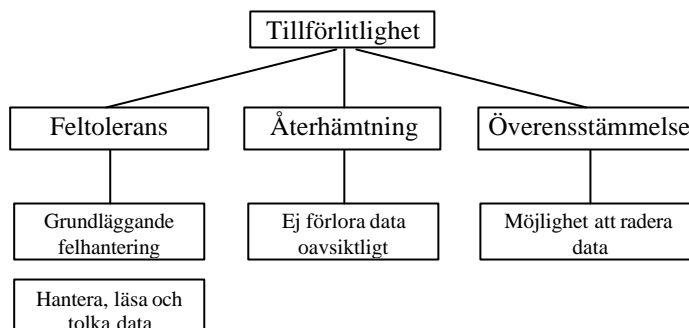
Svårigheten med att designa produktattributen för *funktionalitet* var att få dem lätta att testa och samtidigt ge ett pålitligt värde. Vid design av produktattributen måste ställning tas till om mätningen måste kvantifieras. Ett bra exempel på det här är presentation och insamling av data där resultatet av testet kan variera mellan olika försök.

En annan typ av problematik är de produktattribut som går att testa i flertalet kombinationer. När testpanelen provade att ställa in tidsintervallet för en loggning testades endast ett par kombinationer av start och sluttider. Att de fungerar behöver inte medföra att alla kombinationer kommer att fungera felfritt.

Systemet fick alltså underkänt för egenskapen *funktionalitet*, vilket innebär att systemet som helhet inte kan få godkänt i det här testet. Kravet för att godkänna systemet är att alla produktattributen under *funktionalitet* godkänns. Testpanelen och utvecklarna var ändå övertygade om att de här bristerna utan problem skulle gå att rätta till.

3.3.2 Tillförlitlighet

För att kunna godkänna egenskapen *tillförlitlighet* så måste, enligt SPI 2000, minst ett produktattribut bli godkänt av testpanelen.



Figur 13 Del av testspecifikationen: egenskapen tillförlitlighet.

Den första underegenskapen är *feltolerans*, vilken syftar till att verifiera systemets förmåga att hantera fel på en grundläggande nivå. Testpanelens uppgift bestod i att försöka få systemet att godta felaktiga indata. Felaktiga indata kan delas upp i två kategorier: logiska och syntaktiska. Panelen uppmärksammade att det saknades kontroll för inmatning av MANnr. När testpanelen skulle testa vad som händer när systemet får kontaktproblem med MOBITEX-nätet visade det sig att systemet inte klarade av att hantera det här. Nästa grundkrav som skulle testas var hur systemet hanterar läsning och tolkning av data. Det här testresultatet är beroende av vad testpanelen anser vara godtagbart. Testpanelen ansåg att systemet återger data på ett korrekt sätt.

Nästa underegenskap, *återhämtning*, syftar till att mäta om programmet förlorar data, t.ex. vid onormalt programslut. Testpanelen lyckades inte påvisa att data går förlorad om programmet avslutas på ett onormalt sätt. Nästa underegenskap är *överensstämmelse* för *tillförlitlighet*. Det här avser möjligheten att kunna radera data via operativsystemet. Testpanelen ger systemet godkänt på den här punkten.

Testpanelen godkände alla produktattribut utom två. Det här medför att systemet är godkänt utifrån egenskapen *tillförlitlighet*.

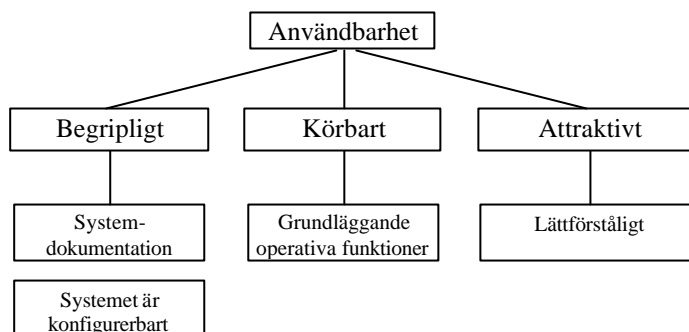
Grundkrav	Produktattribut	J/N
Grundläggande felhantering	Grundläggande felhantering, är indatan syntaktiskt korrekt?	J
	Grundläggande felhantering, är indatan logiskt korrekt?	J
	Hantera kontaktproblem med NCC	J
Hantera, läsa och tolka data	Korrekt återgivning	J
Ej förlora data oavsiktligt	Vid onormalt programavslut skall data ej gå förlorad	J
Möjlighet att radera data	Loggfiler kan raderas från operativsystemet	J

Figur 14 Resultat från testning av tillförlitlighet.

Svårigheten med att designa tester för att prova *tillförlitlighet* är att kraven var mindre konkreta. Det som testpanelen upplevde som svårast var att verifiera den grundläggande felhanteringen.

3.3.3 Användbarhet

Kraven på *användbarhet*, enligt SPI 2000, är de samma som föregående egenskap, d.v.s. minst ett produktattribut måste vara godkänt av testpanelen för att egenskapen skall få godkänt.



Figur 15 Del av testspecifikationen: egenskapen användbarhet.

Den första underegenskapen, *begripligt*, testar om systemdokumentationen och systemets konfigurerbarhet svarar mot kraven. Eftersom systemdokumentationen inte var färdigställd vid tidpunkten för testning kunde denna punkt inte utvärderas av testpanelen och därmed inte godkännas. Systemets konfigurerbarhet godkändes av testpanelen då kartor och basstationer gick att konfigurera.

Den andra underegenskapen, *körbart*, testade systemets grundläggande operativa funktioner. Dels skulle det gå att genomföra en loggning, dels skulle loggningen kunna tolkas visuellt och textenligt. Testpanelen kom fram till att det var genomförbart.

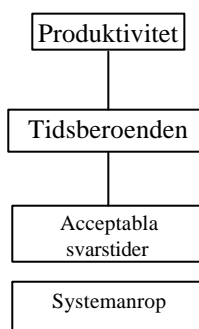
Underegenskapen *attraktivt* syftar till hur förståeliga de grundläggande operativa funktionerna är. Testpanelen upplevde att det var enkelt att starta och avsluta en loggning, då användargränssnitten var intuitiva och utförandet logiskt. Att förstå en loggning textenligt presenterad, bedömdes som enkelt av testpanelen, men felaktig formatering av texten gav attributet underkänt. Enligt testpanelen beror det främst på att det går att se all data från loggningen textenligt samtidigt som det går att se den visuellt som tolkningen av den aktuella loggningen underlättas. Testpanelen upplevde att det även var enkelt att studera en loggning visuellt då inläringströskeln för att förstå presentationen av data var låg. Med det här avsåg de tiden det tar att lära sig förstå beståndsdelarna och dess inbördes relationer i en visuell presentation.

Sammanfattningsvis godkänner testpanelen alla grundkrav för egenskapen *användbarhet* med undantag för systemdokumentationen. Det här medför att egenskapen *användbarhet* är godkänd enligt SPI 2000.

Grundkrav	Produktattribut	J/N
System-dokumentation	Granskas av kund	N
Systemet är konfigurerbart	Konfigurering av kartor	J
	Konfigurering av basar	J
Grundläggande operativa funktioner	Genomföra en loggning (starta/stoppa)	J
	Kunna tolka loggning visuellt	J
	Kunna tolka loggning textenligt	J
Lättförståligt	Förståligt gränssnitt för att starta loggning	J
	Förståligt gränssnitt för att studera loggning textenligt	N
	Förståligt gränssnitt för att studera loggning visuellt	J

Figur 16 Resultat från testning av användbarhet.

3.3.4 Produktivitet



Figur 17 Del av testspecifikationen: egenskapen *produktivitet*.

För egenskapen *produktivitet* är det minst ett produktattribut som skall uppfyllas, för att egenskapen skall få godkänt enligt SPI 2000.

I vår testspecifikation behandlas en underegenskap, *tidsberoenden*, som syftar till att testa systemets svarstider och systemanrop. Kravet är att vid val av grafisk loggpresentation skall den maximala svarstiden vara 30 sekunder. Vidare skall applikationen maximalt belasta MOBITEX-systemet med ett kommando per minut och mobil.

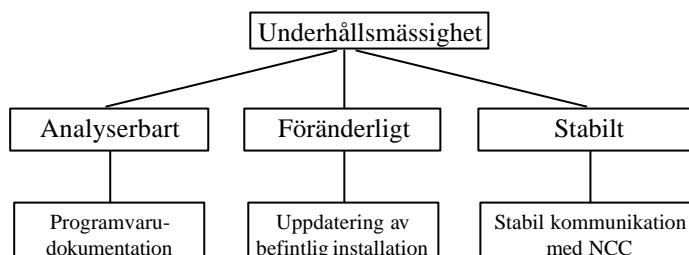
Vissa av produktattributen är beroende av hårdvaran som används och hur systemet är konfigurerat. Det här innebär att dessa produktattribut endast är testade under de förutsättningar som finns uppställda i testspecifikationen. Förutsättningarna speglar de normala driftförhållanden som systemet är ämnat för. Ett exempel på ett produktattribut med sådana beroenden är kravet på den grafiska presentationens maximala svarstid.

Testpanelen uppmätte acceptabla svarstider, men på grund av tidigare fel gick det inte att mäta belastningen. Resultatet blev att de godkände de ett av produktattributen under den här egenskapen. *Produktiviteten* är därmed godkänd enligt SPI 2000.

Grundkrav	Produktattribut	J/N
Acceptabla svarstider	Användarens svarstid vid val av grafisk loggpresentation skall vara maximalt 30 sek.	J
Systemet är konfigurerbart	Maximalt belasta NCC med I genomsnitt ett kommando per minut och mobil	N

Figur 18 Resultat från testning av produktiviteten.

3.3.5 Underhållsmässighet



Figur 19 Del av testspecifikationen: egenskapen underhållsmässighet.

För den här egenskapen gäller som för *produktiviteten* att minst ett av produktattributen skall godkännas för att egenskapen skall godkännas.

För *underhållsmässighet* redovisar vi för tre underegenskaper där grundkraven gäller programvarudokumentation, uppdatering av befintlig installation samt stabil kommunikation med de existerande systemen i MOBITEX-nätet.

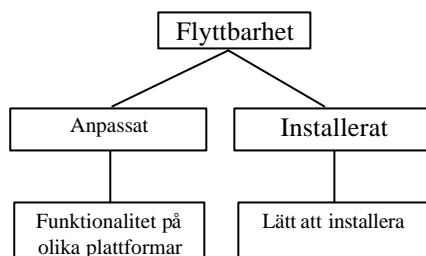
Gällande programvarudokumentationen var inte vid testillfället färdigställd och kunde därför inte godkännas. Programdokumentationen innefattar dels kommenterad kod, dels en översiktlig beskrivning av programstrukturen. Vidare var det heller inte möjligt att godkänna kommunikationen med MOBITEX-nätet.

Testpanelen ansåg att systemets möjligheter till uppdatering var god. Det här medförde att systemet fick godkänt för egenskapen *underhållsmässighet* enligt SPI 2000.

Grundkrav	Produktattribut	J/N
Programvarudokumentation	Kommenterad kod samt översiktlig beskrivning av programstruktur	N
Uppdatering av befintlig installation	Godtagbar uppdateringstid	J
Stabil kommunikation med NCC	Hantera kontaktproblem med NCC	N

Figur 20 Resultat från testning av underhållsmässighet.

3.3.6 Flyttbarhet



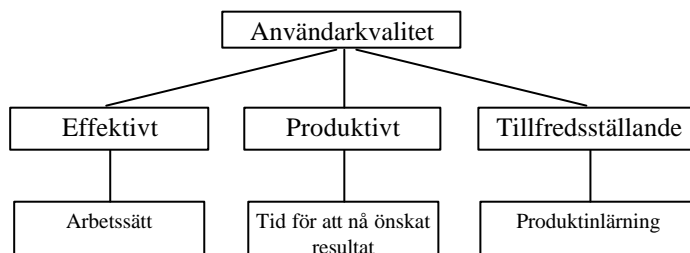
Figur 21 Del av testspecifikationen: egenskapen flyttbarhet.

Enligt testspecifikationen avser *flyttbarhet* att verifiera applikationens funktionalitet under UNIX samt hur lätt den är att installera. Testpanelen godkände de båda produktattributen och därmed godkändes, i enlighet med SPI 2000, *flyttbarhet* för systemet.

Grundkrav	Produktattribut	J/N
Funktionalitet på olika plattformar	Funktionalitet under UNIX	J
Lätt att installera	Installationsmanual	J

Figur 22 Resultat från testningen av *flyttbarhet*

3.3.7 Användarkvalitet



Figur 23 Del av testspecifikationen: egenskapen användarkvalitet.

Till skillnad från de andra egenskaperna är *användarkvalitet* inte mätbar på samma sätt. Egenskapen syftar till att utreda testpanelens känslor och uppfattningar av systemet. Testspecifikationen omfattar tre underregenskaper, *effektivt*, *produktivt* och *tillfredsställande*, som testpanelen har utvärderat.

Grundkravet under *effektivt* är att utreda hur effektivt det går att arbeta med systemet, alltså arbetsättet. Testpanelen uttryckte att systemet är enkelt och tidsbesparande. Systemet presenterar problematiken på ett för testarna logiskt och strukturerat sätt. Den tidsbesparande aspekten visar sig främst genom att användaren kan lägga mer tid på att analysera data då systemet står för insamlingen, vilket användaren tidigare gjorde manuellt.

Den tid som går åt att nå önskat resultat, jämfört med tidigare arbetsätt, har minskat avsevärt, enligt testpanelen. De tyckte att systemets främsta fördelar är att de kan utföra annat arbete parallellt med att systemet samlar in data, vilket var svårt tidigare, samt att endast relevant data presenteras.

Testpanelen tyckte att det var enkelt att komma igång med användandet av systemet. Faktorer som bidrog till det här var bl.a. att de kände igen sig i terminologin samt att det finns få möjligheter att utföra något fel i systemet.

De tre underegenskaperna under *användarkvalitet* godkändes av testarna.

Grundkrav	Produktattribut	J/N
Arbetsätt	Bedömning	J
Tid för att nå önskat resultat	Bedömning	J
Produktinläring	Bedömning	J

Figur 24 Resultat från testning av användarkvalitet.

3.3.8 Sammanfattning av utvärdering

Efter genomförd testning står det klart att systemet inte blev godkänt enligt de normer som SPI 2000 ställer på egendeclaration av programvara. Utvecklare och användare ser inte det här som något oväntat, då det alltid finns buggar i nyutvecklad programvara, och det krävs oftast mer än en testomgång för att programvaran skall godkännas. Det faktum att systemet inte är färdigtestat påverkar inte fallstudiens resultat, då användarna är nöjda med systemets helhetsegenskaper.

4 Diskussion

I det här avsnittet kommer vi att diskutera utifrån resultaten av vår undersökning, så att vi kan dra slutsatser som besvarar vår frågeställning:

Kan den beskrivna utvecklingsmetoden användas för att skapa ett DSS av ett antal existerande system?

Vår slutsats är att vår sammanställda utvecklingsmetod gick att använda i den beskrivna fallstudien för att utveckla ett DSS av existerande system. Vi anser att utvecklingsmetoden går att använda i liknande fall där syftet är att skapa ett DSS av ett antal existerande system.

För att visa det här kommer vi att diskutera utifrån två punkter vilka motiverar svaret på frågeställningen.

?? Har den beskrivna utvecklingsmetoden använts på ett korrekt sätt?

?? Motsvarar systemet ERVs behov?

Om inget annat framgår skall ”vi”, i diskussionen, tolkas som att det är våra åsikter som forskare som framförs.

4.1 Har den beskrivna utvecklingsmetoden använts på ett korrekt sätt?

För att konstatera huruvida den beskrivna utvecklingsmetoden använts på ett korrekt sätt kan vi använda oss utav de framgångsfaktorer som presenterats tidigare (se kap. 1.2.5). Framgångsfaktorerna verifierar om den beskrivna utvecklingsmetoden har använts på ett sätt som är fördelaktigt vid skapandet av ett DSS.

Enligt intervjuerna med användare och beställare har utvecklarna varit omsorgsfulla i sitt tillvägagångssätt och spenderat mycket tid för att sätta sig in i problematiken och tillgodose behoven. Utvecklarna beskriver i dagboken att utvecklingsmetodens aktiviteter sätter användarna i fokus. Testspecifikationen säkerställer att systemet är utformat efter användarnas behov och önskemål, dels då användarna varit delaktiga i utformningen av den, dels då systemet godkänts, i allt väsentligt, i enlighet med specifikationen. Det framgår från alla delar av resultatet att utvecklarna tagit sig tid för att ta fasta på vad användarna verkligen vill ha och att utvecklingsmetoden stödjer det här.

Under utvecklingsarbetet hade utvecklarna ett gruppmöte med användarna som syftade till att kartlägga användarnas kunskaper och deras beslutsprocess. Den här kartläggningen fortsatte under prototypingen även om den inte var lika uttalad. Utvecklarna upplevde att gruppmötet inte gav det resultat som de hade hoppats på men att den ändå var värdefull för det fortsatta arbetet.

Det finns inga resultat från intervjuerna som direkt visar på att användarna tycker att deras kunskaper och beslutsfattande negligerats. Det här beror på att det inte går att fråga användare och beställare om vilka kunskaper de använder för att lösa problem samt att de inte är medvetna om sina beslutsprocesser. Det som går att ta fasta på är att utvecklarna tycker att de på ett bra sätt tagit tillvara användarnas kunskaper och kartlagt deras beslutsprocess.

Utvecklarna uttrycker i dagboken att de har verkat för ett nära samarbete med användarna vilket även styrks av intervjuerna med användare och beställare. Då formerna för samarbetet och till vilken grad användarna bör vara involverade inte är specificerad i utvecklingsmetoden blir konsekvensen att vi inte kan uttala oss om det här samtidigt som vi konstaterar att det här kan vara en brist i utvecklingsmetoden. Utifrån utvecklarnas perspektiv har användarnas behov, utifrån den ursprungliga kravspecifikationen, uppdaterats under konceptualiseringen vilket även användare och beställare bekräftar i intervjuerna. Sammantaget pekar resultaten på att utvecklingsmetoden ger utvecklare ett gott stöd när det gäller förmågan att upprätthålla ett nära samarbete med användarna samt att under hela processen tillgodose de nya behov som dyker upp.

Då resultatet av testspecifikationen visar på att användarna uppfattar systemet som produktivt, effektivt och lättförståeligt anser vi att systemet är enkelt för användarna att använda. Utvecklarna poängterar, i dagboken, att de varit kritiska vid granskandet av nya behov och prioriterat vital funktionalitet. Vi anser att det här bidrar till att behålla enkelheten i systemet.

Utifrån intervjuerna med användaren och beställaren framgår det att de har uppfattat utvecklarnas arbetssätt som strukturerat. Vi anser att det här visar på att det har funnits en övergripande plan på hur utvecklingsprojektet skall bedrivas. Utvecklarna har använt sig av den framtagna utvecklingsmetoden som innefattar evolutionär prototyping. Vi anser därför att metoden som helhet visar på att det finns en tydlig struktur. Den evolutionära prototypingen bidrar till att användarna har lätt för att få med nya idéer och har möjlighet att utforska problematiken i sin egen omgivning. Vi anser även att hela utvecklingsmetoden är evolutionär då den gradvis utökar bilden av problemområdet.

Utifrån det som utvecklarna har gjort ser vi att det har förekommit vissa problem. Ett exempel är gruppmötet som användes i syfte att kartlägga beslutsprocessen. En svårighet låg i att användarna hade olika uppfattningar om problemet. Utvecklarna gjorde en ansats till att ena dem om en gemensam bild. Hade utvecklarna tagit tillvara på de olika uppfattningarna skulle det här kunnat resultera i flera sätt att se på problemet hade kommit fram, vilket är att föredra på det här stadiet.

Det har även kommit fram att det har funnits brister i kommunikationen och då framförallt att den har varit dåligt strukturerad. Utvecklarna har inte dokumenterat förslag till förändringar under konceptualiseringen, vilket har varit ett irritationsmoment samt att vissa förslag kan ha blivit förbisedda. Vidare har användaren uttryckt önskemål om att de vill ha mer dokument om hur projektet fortlöper. Metoden borde alltså tala om vilken dokumentation som skall produceras och när det skall göras.

Strukturen kan förbättras i fler avseenden, t.ex. i kontakten med användaren. När ett DSS utvecklas så går det att se t.ex. genom framgångsfaktorerna att användarens synpunkter och önskemål verkligen måste komma fram i utvecklingsarbetet. Vi tycker att vi lyckats med det här i vårt utvecklingsarbete. Vidare har vi befunnit oss i en situation där användare och beställare har varit intresserade av utvecklingsarbetet vilket har medfört att de har varit lättillgängliga. I normala fall tror vi dels att

användarens engagemang är mindre dels att projektets omfattning är större. Metoden har fungerat för utvecklarna i det här fallet men vi rekommenderar att den skall ta fasta på hur förändringar skall dokumenteras under konceptualiseringsfasen samt hur användarnas medverkan struktureras i utvecklingsarbetet.

Ur ovanstående framgår att utvecklingsmetoden har använts på rätt sätt, men vi har funnit förslag till hur den kan förbättras.

4.2 Motsvarar systemet ERVs behov?

I det föregående avsnittet diskuterade vi om metoden använts på rätt sätt. Följaktligen är det intressant att se till om rätt system har utvecklats. Motsvarar systemet ERVs behov kan vi säga att vi skapat ett DSS av ett antal existerande system.

För att kunna utreda om systemet motsvarar ERVs behov måste följande fastslås:

1. Speglar kravspecifikationen ERVs behov?
2. Motsvarar testspecifikationen de uppsatta kraven?
3. Är systemet godkänt enligt testspecifikationen?

För att svara på första frågan kommer vi dels att se till hur utvecklarna med hjälp av utvecklingsmetoden har tillgodosett ERVs behov, dels hur användarna och beställaren har upplevt det här. Utvecklarna har arbetat nära med användarna under hela utvecklingsprocessen. Under de tre första faserna har det här samarbetet varit intensivare för att bättre kunna utreda ERVs behov och sammanställa dem i kravspecifikationen.

Planeringsfasen har syftat till att förbereda användarna och beställaren för det kommande utvecklingsarbetet genom att få dem att se på problemområdet ur ett vidare perspektiv. Det vidare perspektivet leder till att utvecklarna lättare kan identifiera de delar av beslutsprocessen som användare och beställare inte själva är medvetna om. I nästa fas, förstudien, var målet att kartlägga delar av beslutsprocessen. Innan det här skedde tog utvecklarna reda på vilka olika roller som existerade inom problemområdet samt utredde obekanta och centrala begrepp i verksamheten. Genom att göra det här minskar risken för att utvecklarna och de övriga inblandade skall missförstå varandra under kartläggningen av beslutsprocessen. Eftersom utvecklarna fokuserar på beslutsprocessen så bidrar det till att ERVs behov sätts i centrum.

Genom att se till kartläggningen av beslutsprocessen och den övergripande beskrivningen av problemområdet kan utvecklarna ta fram rika bilder. Dessa speglar dels hur utvecklarna uppfattar problemsituationen dels ett första förslag till lösning. Rika bilder är, i det här fallet, att betrakta som en kvittens på att utvecklarna har en uppfattning av problemsituationen som överensstämmer med användare och beställare.

Målet med systemdefinitionen är att på ett kortfattat sätt beskriva vad systemet skall omfatta. Den här beskrivningen skall inkludera de aspekter som är relevanta för att utveckla ett system. I det här arbetet är VATOFF ett viktigt verktyg för att validera systemdefinitionen. Systemdefinitionen är ytterligare ett sätt att kontrollera att

utvecklarna har tagit fram rätt förslag till lösning på användaren och beställarens problem.

Utifrån den valda systemdefinitionen sätts de funktionella kraven som systemet skall uppfylla. Kraven formar tillsammans med systemdefinitionen kravspecifikationen, som även tar hänsyn till den teknik som systemet skall baseras på och i vilken miljö det skall utvecklas i.

Enligt användaren och beställaren har utvecklarna visat intresse för deras behov och önskemål, vilket de anser framgår i kravspecifikationen. ERV har alltså själva godkänt kravspecifikationen. Med anledning av det här och föregående resonemang anser vi att kravspecifikationen speglar ERVs behov.

För att ta reda på om testspecifikationen motsvarar de uppsatta kraven kommer vi att se till hur den har tagits fram. För att utforma testspecifikationen utgick utvecklarna från vart och ett av kraven och bröt ner dem i mindre mätbara enheter. För att ett krav skall godkännas måste alla delarna uppvisa godtagbara mätvärden under testningen. Utöver kraven syftar testspecifikationen till att mäta användarkvalitet, vilket bidrar till att systemet verkligen är användbart.

Beställaren och användarna har varit med under framtagandet av testspecifikationen och även varit med om att godkänna den. Vi anser alltså att testspecifikationen motsvarar de uppsatta kraven.

Slutligen, är det utvecklade systemet godkänt enligt testspecifikationen? Utvecklarna har tillsammans med användare testat systemet och har funnit att systemet är snudd på godkänt enligt testspecifikationen.

Trots att det mesta pekar på att systemet motsvarar ERVs behov så ser vi en del tvetydigheter. För det första upplever vi att utvecklarna kunde ha gjort mer för att kartlägga beslutsprocessen om utvecklingsmetoden explicit hade beskrivit hur det åstadkoms.

För det andra ifrågasätter vi om VATOFF är den mest lämpliga metoden för att validera systemdefinitionen vid utvecklandet av ett DSS, och speciellt i vår fallstudie. Utvecklarna borde undersökt om det fanns någon tillgänglig metod för att validera en systemdefinition av ett DSS. Ett annat alternativ är att de kunde anpassat den använda valideringsmetoden till att omfatta specifika aspekter som tar hänsyn till framtagandet av ett DSS.

Vi har utrett att ERVs behov framkommer i kravspecifikationen och att testspecifikation motsvarar de här kraven. ERV har godkänt både specifikationerna och systemet som helhet, vilket framkommit utifrån testet av systemet. Därför anser vi att systemet motsvarar ERVs behov, vilket är ett DSS baserat på ett antal existerande system. Systemet underlättar insamling, bearbetning och presentation av data samt avspeglar användarnas begreppsvärld, värderingar och verklighetsuppfattning på ett mer överskådligt sätt. Genom systemet kan användaren nu tillgodogöra sig väsentlig information på kortare tid.

4.3 Finns det yttre faktorer som påverkat utvecklingsmetodens resultat?

För att validera vårt arbete som forskare skall vi belysa de yttre faktorer som vi anser kan ha påverkat utgången av fallstudien.

För det första är det viktigt att beakta att användarna och beställaren är professionella systemutvecklare och verifierare. Det här medför att de är väl medvetna om vad vi är ute efter i utvecklingsarbetet. Det bidrar även till att det blir avsevärt lättare att kommunicera med dem samtidigt som de kan ha påverkat utvecklarna i en felaktig riktning.

Den andra faktorn är projektbeskrivningen, som fanns tillgänglig för utvecklarna från projektets start. Det faktum att det fanns en projektbeskrivning tyder på att ERV redan på förhand hade skapat sig en uppfattning om vad problemet var, samt en del funderingar kring hur problemet kunde lösas. Vi undrar om det här påverkat utvecklarnas objektiva omdöme.

För det tredje undrar vi om vår situation, som utvecklare och forskare, har påverkats av det faktum att vi har varit på ERV för att utföra ett examensarbete/skriva en magisteruppsats. Det här kan ha bidragit till att användare och beställare har spenderat extra tid med oss samt visat extra intresse för vårt arbete.

4.4 Vidare forskning

Vi har under vårt arbete fått några uppslag till hur vidare forskning i området skulle kunna bedrivas. Den här forskningen syftar till att förfina utvecklingsmetoden.

Ett förslag till förbättringar i metoden är, som tidigare nämnts, se om det finns en bättre metod för att validera systemdefinitionen som är mer inriktad på utvecklingen av DSS. Ytterligare ett förslag till förbättring är att använda en metod som underlättar kartläggandet av beslutsprocessen.

Vi har avgränsat vår fallstudie till att omfatta steg ett till fem i vår utvecklingsmetod. Det hade varit intressant att se om ett fullständigt genomförande hade förändrat resultatet. Vidare hade det även varit intressant att se till de långsiktiga effekterna av systemet.

Källförteckning:

Böcker:

Brown, David, 1997,

An Introduction to Object-Oriented Analysis, Objects in Plain English,
John Wiley & Sons, Inc.

Easterby-Smith, Mark et al., 1991,

Management Research – An Introduction,
SAGE Publications.

Eriksson, Lars Torsten och **Wiedersheim-Paul, Finn**, 1997,

Att utreda, forska och rapportera,
Liber Ekonomi, Malmö, Sverige.

Finlay, Paul, 1994,

Introducing Decision Support Systems,
NCC Blackwell Ltd.

Finne, Thomas, 1998,

A Decision Support System for Improving Information Security,
Turku Centre for Computer Science, TUCS Dissertations, Åbo, Finland.

Langefors, Börje, 1995.

*Information Systems Theory and Infology. Essays on Infology. Summing up and
Planning for the Future*,
Gothenburg Studies in Information Systems, Göteborg, Sverige.

Lewis, Paul, 1994,

Information Systems Development,
Pitman Publishing, London, England.

Mathiassen, Lars, et al 1998,

Objektorienterad analys och design,
Studentlitteratur.

Sommerville, Ian, 2001

Software Engineering,
Addison-Wesley.

Sprague, Ralph H. och **Watson, Hugh J.**, 1993,

Decision Support Systems, Putting Theory into Practice,
Prentice Hall International, Inc.

Turban, Efraim och **Aronson, Jay E.**, 1995,

Decision Support and Expert Systems: Management Support Systems,
Prentice Hall International, London, England.

Rapporter:

Håkansson, Bertil 2000 *Egendeklaration av programvaror – Svensk metod för kvalitetsmärkning av programvaror med internationella standarder som grund.*
ISSN 0283-5266

Webbsidor:

Joiner, David "A Summary of Principles for User-Interface Design"

Publiceringsdatum: 1998-08-14, inhämtat 2001-04-25

http://www.sylvantech.com/~talin/projects/ui_design.html

Lotsson, Anders *De kritiska sekunderna som skapar informationsstress*

Publiceringsdatum: 1998-12-02, inhämtat 2001-05-17

<http://domino.idg.se/cs/artikel.nsf/674b84618b948c0cc12567d20050feb7/f01ad60ecd53abddc12566ce00283e29?OpenDocument>

Ericsson *Informationssida MOBITEX*

Publiceringsdatum: 2000-10-30, inhämtat 2001-05-17

<http://www.ericsson.com/wireless/products/mobsys/mobitex/about.shtml>

Att skapa beslutsstödssystem från existerande system genom att använda Soft Systems Methodology och Prototyping

Bilagor

BILAGA 1, PROJEKTSPECIFIKATION.....
BILAGA 2, INTERVJUGUIDE.....
BILAGA 3, TESTSPECIFIKATION NIVÅBESKRIVNINGAR.....
BILAGA 4, KRAVSPECIFIKATION "MOPS".....
BILAGA 5, TESTSPECIFIKATION.....
BILAGA 6, TIDSPLAN "MOPS"
BILAGA 7, RIK BILD.....
BILAGA 8, GRÄNSSNITT I "MOPS".....

Bilaga 1, Projektbeskrivning

Mobilpositionering i MOBITEX

Beskrivning

MOBITEX är ett mobildatasystem för bl.a. mobil internetaccess. Systemet finns utbyggt i många länder världen över och baserar sig på teknologi från Ericsson. Sektionen System Verification ansvarar bl.a. för driftsupport till en svensk operatör.

Vid drift av ett mobildatanät kan det vara av intresse att veta var de mobila terminalerna befinner sig någonstans. Idag finns denna information tillgänglig i realtid och kan presenteras i ett teckenbaserat gränssnitt. Syftet med exjobbet är att designa och implementera en applikation i mjukvara där en eller flera mobila terminalers position kan visas grafiskt. Information skall förslagsvis presenteras i ett webb-baserat användargränssnitt. En applikation som denna kommer att ge oss förbättrade möjligheter att supporta våra kunder.

Examensarbetet kommer att inledas med en analysfas där kraven på verktyget definieras. Vidare skall funktionalitet implementeras i mjukvara. I arbetet ingår också att ta fram användardokumentation för framtagna applikation. Denna skall också testas under verklighetsnära betingelser. Det finns mycket goda möjligheter att testa det framtagna verktyget i ett "skarpt" MOBITEX-nät som sektionen förfogar över. Slutligen skall resultatet dokumenteras och presenteras.

Omfattning

- ?? Analys av kravbild.
- ?? Kravspecifikation av applikation.
- ?? Design och implementering i mjukvara.
- ?? Genomföra praktiska tester.
- ?? Redovisning av resultat muntligt och skriftligt.

Examensarbetet utföres lämpligen av två till tre personer från utbildningslinje inriktad mot telekom och data. Omfattningen är anpassad mot 20 poäng.

Bilaga 2, Intervjuguide

Kommunikation

Anser ni att det har varit lätt att få kontakt med oss som utvecklare?

Har du upplevt att det har varit någon skillnad under utvecklingens gång att få dina idéer uppmärksammade? (Har du sett olika faser?)

Tycker ni att ni har fått tillräckligt med information av oss för att ha förståelse för det arbete som vi bedrivit?

Tycker ni att vår kunskap om tekniken (MOBITEX) har varit tillräcklig för att kunna föra en fruktsam diskussion?

Inflytande

Anser du att dina åsikter och krav återspeglas i kravspecifikationen?

Tycker ni att vi har tagit åt oss tillräckligt mycket av era önskemål och krav?
(Har vi gjort tillräckligt för att få fram kraven??)

Anser du att något viktigt krav, från din sida, blivit förbisett?

Har du upplevt att du varit tvungen att vara påstridig för att få igenom dina önskemål?

Samarbete

Har ni upplevt att det har funnits några barriärer mellan er som beställare och oss som utvecklare?

Hur tror ni att det hade påverkat resultatet att ta hjälp av oss som utomstående utvecklare (till skillnad från att göra det själva)?

Vad tror ni det hade blivit för skillnad ifall någon från avdelningen hade varit med oss under implementeringen?

Kvalitet

Anser du att vi har gjort rätt saker och i en rimlig ordning?
(Har vi gjort de saker som ni ville att vi skulle göra?)

Anser du att vi utfört arbetet på rätt sätt?
(Har vi gjort de saker vi utfört på ett korrekt sätt?)

Är det något som ni anser att vi borde gjort på ett annorlunda sätt?

Vad har du för uppfattning om strukturen i vårt sätt att arbeta?

Övrigt

Hade du en klar bild av vad ni ville ha innan vi påbörjade vårt arbete?

Tror ni att det hade varit möjligt att definiera hela systemet på papper från början?

Om du jämför vårt sätt att arbeta med andra exjobbare som ni har haft, tycker du att det skiljer sig och är det i så fall en positiv eller negativ skillnad?

Tycker du att vårt sätt att jobba har varit alltför fokuserat på den teknik och de verktyg som skall användas?

Bortsett från det som redan framkommit, vad anser du är bra kriterier för kvalitet i ett utvecklingsprojekt?

Bilaga 3, Testspecifikation nivåbeskrivningar

Niva D Liten skada på egendom, ingen risk för människor.
Niva C Skada på egendom. Få människor skadade.
Niva B Hot om människors liv.
Niva A Många människor dödade.

	Nivå D	Nivå C	Nivå B	Nivå A
Funktion	Funktionell provning (Black Box)	Granskning (Checklista)	Komponent-provning (Glass box)	Formellt bevis
Tillförlitlighet	Programmerings språk hjälpmedel	Analys av feltoleransen	Pålitlighet tillväxt-modell	Formellt bevis
Användbarhet	Inspektion av gränssnitt	Överensstämmelse mot gränssnittets standard	Provning i laboratorium	Användarens mentala modell
Effektivitet	Mätning av exekveringstid	Benchmarkning-testning	Algoritmisk komplexitet	Analys av utförande-profil
Underhållbarhet	Granskning av dokument (Checklistor)	Statisk analys	Analys av utvecklings-processen	Utvärdering av spårbarhet
Flyttbarhet	Analys av installationen	Överensstämmelse mot programmerings-regler	Utvärdering av omgivningens begränsning-ar	Utvärdering av program-planering

Bilaga 4, Kravspecifikation "MOPS"

Introduktion

Det här är kravspecifikationen för *MOPS* – Mobil Positionering och Status, ett examensarbete på Ericsson Mobile Data Design (ERV/M/PV). *MOPS* är en applikation för att presentera mobilers geografiska position och status. Applikationen utvecklas som ett led i vår magisteruppsats vid institutionen för Informatik, Göteborgs Universitet. Målet med examensarbetet är att implementera kraven för prioritet ett. Kraven under prioritet två utförs i mån av tid.

Problemformulering

För att undersöka MOBITEX-nätets status används ett antal olika applikationer. De här är delsystem i NCC. Det finns exempelvis olika system för att rapportera larm, hämta information om mobilers status och position, och undersöka BASARnas momentana belastning. För att skapa sig en övergripande bild över nätets driftstatus, exempelvis vid larm, måste operatören sammanställa information från ett antal olika delsystem vilket är tidskrävande och krångligt. Nätoperatören får svårt att skapa sig en överblick, det är exempelvis komplext att se var nätets brister finns, t.ex. i täckning eller i kapacitet. För en operatör med hög teknisk kompetens rörande MOBITEX är det möjligt att sköta driften av ett nät med dessa verktyg. För en operatör med lägre MOBITEX-kompetens kan samlad information kombinerat med en grafisk presentation underlätta handhavandet av nätet.

Systemdefinition

Systemet skall grafiskt visualisera positionering av ett antal mobila enheter i ett MOBITEX-nät samt förmedla information om de BASAR mobilen befinner sig vid. Systemet skall visualisera positionering av mobila enheter i ett MOBITEX-nät för att kunna utreda orsaken till problem, som uppstår i nätet. Det skall gå att logga en eller flera mobila enheter under ett givet tidsintervall. Loggningen skall kunna presenteras grafiskt dels för ett givet tidsintervall, dels för en tidpunkt. Systemets utdata skall primärt vara beslutsunderlag för operatörens dagliga arbete men skall även kunna sammanställa statistiska rapporter. Information om BASAR loggas då den mobila enheten är aktiv under BASen. BASAR skall även kunna loggas separat, likt de mobila enheterna, på användarens initiativ. Syftet med det här är att få en bättre helhetsbild över problemen som kan uppstå i nätet. Det underlättar att samla information om både BASAR och mobila enheter i en applikation. Det sammantagna värdet av informationen är större än delmängden d.v.s. informationen om mobilerna respektive BASARna för sig.

Användarna skall ha grundläggande kunskaper i MOBITEX uppbyggnad. Systemet skall användas och utvecklas i en Unix-miljö.

VATOFF

Villkor:

Användaren är operatör av ett MOBITEX-system, med skiftande teknisk kunskap om systemets uppbyggnad. Systemet är en prototyp för fortsatt utveckling. Ericsson Mobile Data Design ansvarar för vidare utveckling av systemet.

Användningsområde:

Övervakning och underhåll av MOBITEX-systemet hos eller för en operatör. Applikationen skall underlätta att lösa problem med MOBITEX-nätet.

Teknologi:

Systemet skall utvecklas och användas i UNIX-miljö, med uppkoppling mot det berörda MOBITEX-nätet. Systemversion NTE.

Objektsystem:

Grafisk visualisering av ett antal mobila enheterna i ett nät samt att presentera information om BASAR för bättre överblick. Att komplettera med information om BASAR leder till en mer samlad bild av problemområdet.

Funktionalitet:

Systemet skall tillhandahålla beslutsunderlag för operatörens dagliga arbete. Applikationen skall underlätta övervakning och utredning av problem i MOBITEX-nätet genom följande:

1. Loggning av data om valda mobila enheter.
2. Loggning av data om valda BASAR.
3. Att visa information om en BAS där vald mobil är aktiv eller senast var aktiv (inroamad).
4. Att grafiskt presentera information för att lättare förstå problemet, presenterar det ur ett annat perspektiv.

Filosofi:

Övervakningsverktyg med tydliga inslag av automatisering. En tilläggsapplikation för att på ett mer användarvänligt och överskådligt sätt kunna övervaka och utreda problem i ett MOBITEX-nät.

Helhetsegenskaper hos systemet

?? *Användarvänlighet.* Grafiskt istället för teckenbaserat.

?? *Överskådligt.* Grafisk karta istället för att själv tolka information om hur mobila enheter rör sig mellan BASAR. Att sammanföra flera informationskällor till en applikation.

Prototypkrav/avgränsningar: prioritet ett

≪≪ Ska kunna välja **antal** mobiler (max 10 st) som skall spåras.

≪≪ De mobiler som har valts loggas under ställbar **tid** (exempelvis var 30:e minut).

≪≪ För varje mobil enhet **logg**as en rad där information om vilken BAS som används, status, datum, tid etc. skall lagras.

≪≪ Mobilernas väg genom nätet skall visas **grafiskt**, eventuellt som ett streck på en karta som visar hur den mobila enheten rört sig i nätet (alltså över vilka BASAR). Det grafiska gränssnittet skall kunna visa ett område på ungefär 15 x 10 mil.

- ☞☞ **Kartan** kommer att vara ganska grov med prickar som motsvarar BASAR och mobila enheter. Det skall gå att **zooma** in ett mindre område för att få reda på hur en eller ett par enheter har rört sig.
- ☞☞ Av de antal mobiler som loggas skall **en eller flera** kunna **väljas** för att visa en grafisk översikt över hur den eller de har rört sig i nätet.
- ☞☞ Det skall gå att **välja** ut en viss **tidsperiod** av den totala loggningen och presentera denna.
- ☞☞ Applikationen skall kunna visa var en mobil enhet befinner sig i **nuläget**.
- ☞☞ Applikationen får **maximalt** belasta nätet med **ett kommando per minut**.

Prototypkrav/avgränsningar prioritet två

- ☞☞ Att kunna välja en BAS som en mobil passerat för att se **BASens kapacitetsutnyttjande** vid det tillfället.
- ☞☞ Att **utifrån en BAS** kunna se **hur många mobila enheter** som utnyttjar den och hur många mobila enheter som enligt BASen är aktiva.
- ☞☞ Ska kunna avläsa hur stor procent av **paketerna som skickas om** för en specifik **BAS**.

Bilaga 5, Testspecifikation

Förutsättningar

- ?? NTE är den release av MOBITEX som skall användas.
- ?? Systemet, MOPS, skall köras på en UNIX (motsvarande Sparc 5) med SunOS 4.X.
- ?? Det skall finnas en grundkarta och fyra specialkartor tillgängliga över göteborgsregionen.
- ?? 6 basstationer skall vara konfigurerade.

Krav på testerna

- ?? Testerna skall utföras av tre användare under övervakning av testledare.
- ?? För att produkten skall bedömas som godkänd måste **samtliga** produktattribut för *funktion* vara uppfyllda samt **minst** ett produktattribut från var och en av de övriga egenskapsgrupperna.

Egenskap	Underegenskap	Grundkrav	Verifikat	Produktattribut
1 Funktion	1.1 Ändamålsenlig	1.1.1 Hantera grundfunktioner	1.1.1.1	Logga mobil (Starta/stoppa)
			1.1.1.2	Välja tidsperiod för loggning. För en given tidsperiod med specificerad start och stopptid. För en ospecificerad tidsperiod, oändlig.
			1.1.1.3	Välja antal mobiler för loggning. Minst en, max tio.
			1.1.1.4	Välja pollintervall. Minst en minut och maximalt 30 minuters intervall.
			1.1.1.5	Visa loggning för mobil. Går det att titta på den loggfil som skapas, under en pågående loggning?
			1.1.1.6	Presentera loggning på karta. Går det att få en grafisk presentation av loggningen på en av användaren specificerad karta?

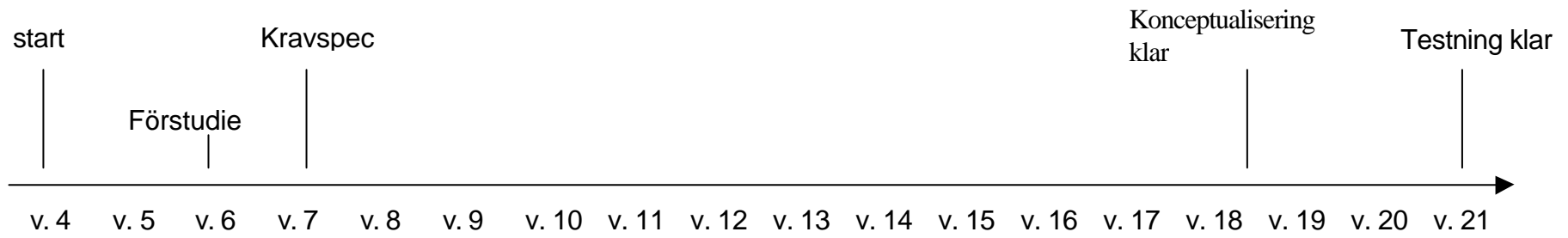
Bilaga 5

			1.1.1.7	Presentera alla mobilers position för en specificerad tidpunkt.
			1.1.1.8	Presentera en mobil för ett valt tidsintervall.
			1.1.1.9	Visa alla mobilers status.
			1.1.1.10	Zoomfunktion för karta
	1.2 Noggrannt	1.2.1 Presentation av data	1.2.1.1	Basstation ska ritas ut med en noggranhet av 1 km
		1.2.2 Insamling av data	1.2.2.1	Maximal avvikelse från pollintervallet får vara 5 min
			1.2.2.2	Maximal avvikelse från valt tidsintervall får vara 10min.
		1.2.3 Lagring av data	1.2.3.1	Mannr, BASnr, mobilstatus, datum och tid ska lagras vid loggning.
	1.3 Samverkan	1.3.1 Hantera system- och användargränssnitt	1.3.1.1	Maximalt belasta NCC med i genomsnitt ett kommando per minut och mobil
	1.5 Överensstämmelse	1.5.1 Hantera MOBITEX-release R14 och NTE		Se FÖRUTSÄTTNINGAR
		1.5.2 Hantera användar-definierade kartor i GIF-format		Se FÖRUTSÄTTNINGAR
2 Tillförlitlighet	2.2 Feltolerans	2.2.1 Grundläggande felhantering	2.2.1.1	Grundläggande felhantering. Är indata syntaktiskt korrekt?
			2.2.1.2	Grundläggande felhantering. Är indata logiskt korrekt?
			2.2.1.2	Hantera kontaktproblem med NCC. Vid förlust av kontakt skall 10 återkopplingsförsök göras därefter avbryts loggningen.
		2.2.2 Hantera, läsa och tolka data	2.2.2.1	Korrekt återgivning
	2.3 Återhämtning	2.3.1 Ej förlora data oavsiktligt	2.2.3.1	Vid onormalt programavslut skall data ej gå förlorad
	2.4 Överensstämmelse	2.4.1 Möjlighet att radera data	2.4.1.1	Loggfiler kan raderas från operativsystem
3 Användbarhet	3.1 Begripligt	3.1.1 Systemdokumentation	3.1.1.1	Granskas av kund.
		3.1.2 Systemet är konfigurerbart	3.1.2.1	Konfigurering av kartor
			3.1.2.2	Konfigurering av BASAR
	3.3 Körbart	3.3.1 Grundläggande operativa	3.3.1.1	Genomföra en loggning (Starta / Stoppa)

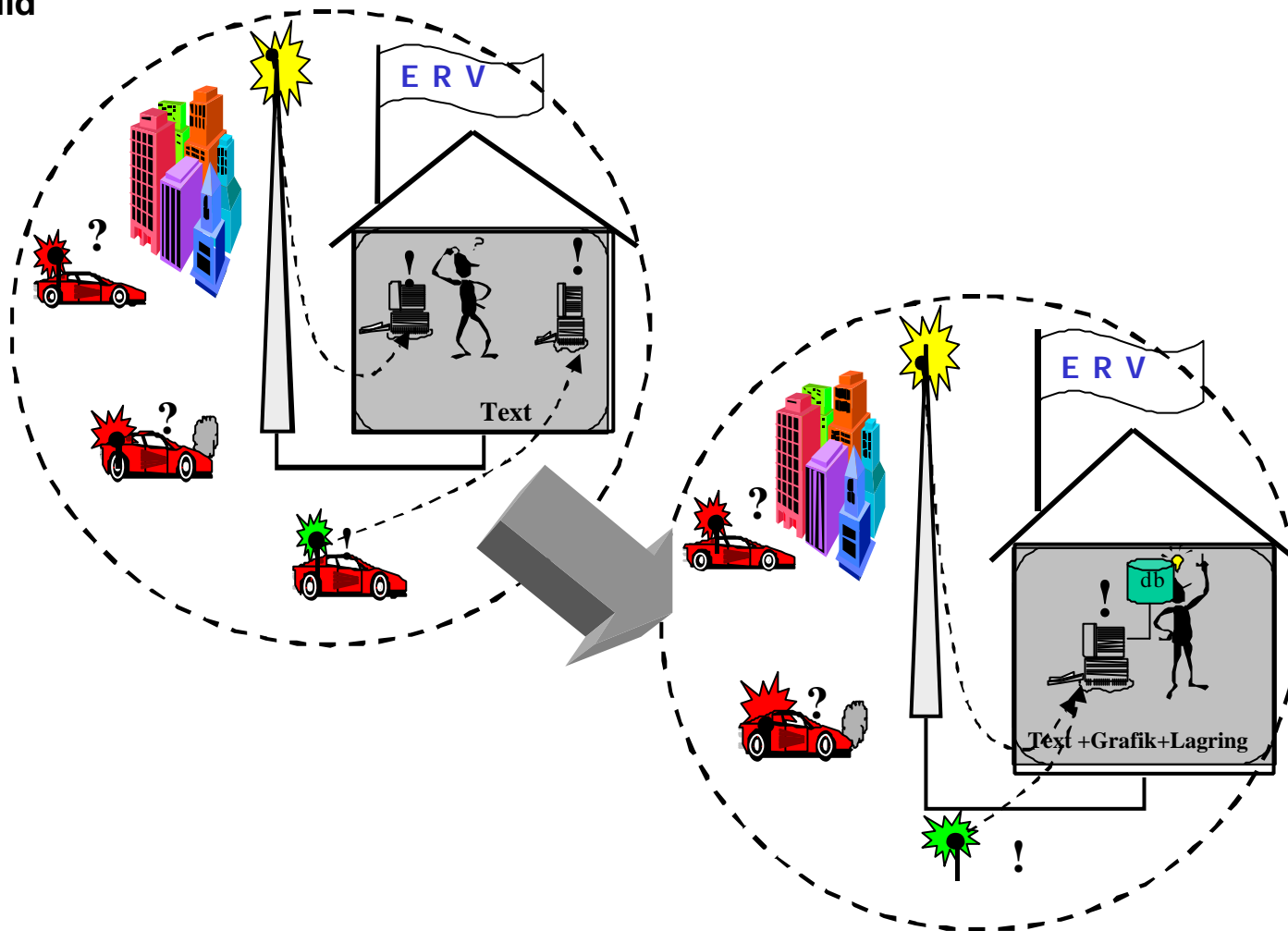
Bilaga 5

		funktioner		
			3.3.1.2	Kunna tolka loggning visuellt
			3.3.1.3	Kunna tolka loggning textenligt
	3.4 Attraktivt	3.4.1 Lättförståligt	3.4.1.1	Förståeligt gränssnitt för att starta loggning
			3.4.1.2	Förståeligt gränssnitt för att studera loggning textenligt
			3.4.1.3	Förståeligt gränssnitt för att studera loggning visuellt
4. Produktivitet	4.1 Tids beroenden	4.1.1 Acceptabla svarstider	4.1.1.1	Användarens svarstid vid val av grafisk loggpresentation skall vara maximalt 30 sek.
		4.1.2 Systemanrop	4.1.2.1	Maximalt belasta NCC med i genomsnitt ett kommando per minut och mobil
5. Underhållsmässighet	5.1 Analyserbart	5.1.1 Programvarudokumentation	5.1.1.1	Kommenterad kod samt översiktlig beskrivning av programstruktur
	5.2 Föränderligt	5.2.1 Uppdatering av befintlig installation	5.2.1.1	Godtaglig uppdateringstid. Systemets uppbyggnad skall tillåta uppdateringar.
	5.3 Stabilt	5.3.1 Stabil kommunikation med NCC	5.3.1.1	Hantera kontaktproblem med NCC. Vid förlust av kontakt skall 10 återkopplingsförsök göras därefter avbryts loggningen.
6. Flyttbarhet	6.1 Anpassat	6.1.1 Funktionalitet på olika plattformar	6.1.1.1	Funktionalitet under UNIX.
	6.2 Installerbart	6.2.1 Lätt att installera	6.2.1.1	Installationsmanual.
7. Användarkvalitet	7.1 Effektivt	7.1.1 Arbetssätt		
	7.2 Produktivt	7.2.1 Tid för att nå önskat resultat		
	7.4 Tillfredsställande	7.4.1 Produktinläring		

Bilaga 6, Tidsplan "MOPS"



Bilaga 7, Rik bild



Bilaga 8, Gränssnitt i "MOPS"

MAN-number

1. 123456
2. 123457
3. 123458
4. 123459
5.
6.
7.
8.
9.
10.
Time-interval

Start: hh mm date
04 10 2001-05-14

End: hh mm date
04 00 2001-05-15

Poll-interval
Every 10:th min

MOBITEX net-revision
NTE

File: Start Clear

Gränssnitt för start av loggning.

MAN NUMBER

110112

Progress

FINISHED

Start date
2001-05-07

End date
2001-05-07

Poll-interval
Every 1:th minute

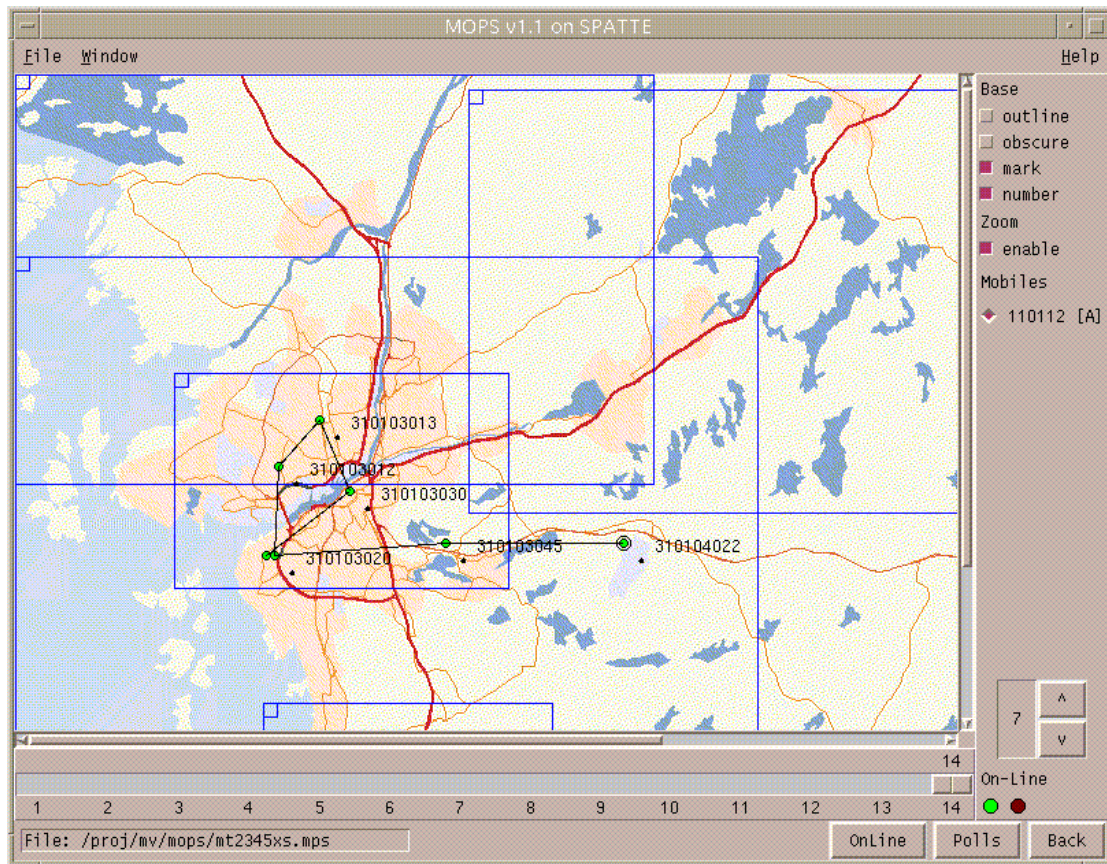
Status
Inactive

POLLS

110112	310190040	2001-05-07	19:12:06	310103020	active	1
110112	310190040	2001-05-07	19:13:14	310103020	active	2
110112	310190040	2001-05-07	19:14:24	310103020	active	3
110112	310190040	2001-05-07	19:15:34	310103020	active	4
110112	310190040	2001-05-07	19:16:44	310103020	active	5
110112	310190040	2001-05-07	19:17:55	310103020	active	6
110112	310190040	2001-05-07	19:19:04	310103020	active	7
110112	310190040	2001-05-07	19:20:14	310103020	active	8
110112	310190040	2001-05-07	19:21:24	310103020	active	9
110112	310190040	2001-05-07	19:22:34	310103020	active	10

File: /proj/mv/mops/prototyp/niklas.mps Map Abort

Gränssnitt för textbaserad presentation av loggning.



Gränssnitt för visuell presentation av loggning.

