

ATT SAMLA IN OCH ANALYSERA KUNDKRAV - En fallstudie i projektet COMPASS

Abstrakt

Användarna på SKF har tröttnat på de gamla systemen med AS/400 och dess tråkiga användargränssnitt. De vill ha PC-Interface istället, med Windows funktionalitet, men fortfarande dra nytta av den funktionalitet som finns på AS/400. SKF har investerat mycket i AS/400 och dess stora fördelar är bl a säkerhet, stabilitet och driftsäkerhet. Hur skall man behandla de kundkraven för det nya systemet som ska ha Windows funktionalitet och samtidigt dra nytta AS/400-funktionaliteten i det gamla systemet? Därför startades projektet Compass som ska bli ett system som uppfyller de nya kraven. Och hur skall dokumentationen tas fram för att anpassas till de nya kraven på systemet? Vilka uppgifter måste komma in och hur skall de dokumenteras? Det måste vara på ett sätt som kunden godkänner och som systemutvecklarna kan arbeta efter. Resultatet av den här undersökningen var att den metodik som man utvecklar system efter idag, ISDM, räcker inte för de nya kraven. Den behöver uppdateras och ytterligare standards arbetas fram.

Jessica Engström
Examensarbete I 10p IA5840
ADB-programmet 80p
Vårterminen 1998

Handledare: Birgitta Ahlbom
Fackhandledare: Ann-Christine Hagberg

INNEHÅLLSFÖRTECKNING

1 INLEDNING	3
1.1 BAKGRUND.....	4
1.2 PROBLEMMOMRÅDE	5
1.3 PROBLEMFÖRMULERING	5
1.4 SYFTE	6
1.5 AVGRÄNSNINGAR	6
1.6 METOD	6
1.6.1 Varför valde jag fallstudie?.....	6
2 BESKRIVNING AV KUNDEN, PROJEKTET OCH METODIKEN	7
2.1 VPO	7
2.2 COMPASS	7
2.3 HUR TAS DOKUMENTATIONEN FRAM IDAG?.....	8
2.3.1 Mål och syften med ISDM	8
2.3.2 Fem användningsområden	8
2.3.3 Allmänna användningsområden	9
2.3.4 Vilka berörs av ISDM?.....	9
2.3.5 Projekt där ISDM är användbar	9
2.3.6 Uppgifter där ISDM är användbar.....	9
3 PROJEKTÖVERSIKT, DE OLIKA FASERNA ENLIGT ISDM	10
3.1 PROJECT INITIATION.....	10
3.2 PRE-STUDY PHASE.....	11
3.3 DEFINITION PHASE.....	12
3.4 REALIZATION PHASE.....	13
3.5 IMPLEMENTATION PHASE.....	14
3.6 OPERATION & MAINTENANCE.....	14
3.7 SAMMANFATTNING PROJECT PHASES	15
3.7.1 Projekt beslut.....	16
4 MAIN ACTIVITIES	17
4.1 PLANNING (01)	17
4.2 ASSIGNMENT DIALOGUE (02).....	18
4.3 FOLLOW-UP AND CONTROL (03).....	19
4.4 COMPILER PROGRAM SPECIFICATION (13).....	19
4.5 DEFINE OBJECTIVES AND CONSTRAINTS (25)	21
4.6 SYSTEM FUNCTIONS AND STRUCTURE (40).....	22
4.7 DEFINE BUSINESS FUNCTIONS & PROCEDURES (45).....	23
4.8 DESIGN SCREEN AND DIALOG (47).....	24
4.8.1 Design principer	24
4.8.2 Krav på skärmlayouten	24
4.8.3 Krav på dialogen	25
4.8.4 Skärmdisposition och layout	25

4.8.4.1 Inmatning	25
4.8.4.2 Urval	26
5 TILLVÄGAGÅNGSSÄTT	27
5.1 NULÄGESANALYS	27
5.1.1 <i>Beskrivning av de inrapporterade problemen</i>	28
5.1.1.1 Problemlista.....	28
5.1.1.2 Mållista.....	29
5.1.1.3 Förändringsbehov	30
5.1.1.4 Åtgärdslista	31
5.2 OBJEKTORIENTERAD VERKSAMHETSANALYS	31
5.3 GRAFER	32
5.3.1 <i>Problemgraf</i>	32
5.3.2 <i>Målgraf</i>	33
5.4 ARBETSPROCESSEN	33
5.5 DATABASDESIGN	33
5.6 JÄMFÖRELSE MELLAN VÅRT ARBETSSÄTT OCH ISDM.....	34
6 VAD ÄR SYSTEMDOKUMENTATIONSKVALITÉ?	35
6.1 INLEDNING.....	35
6.2 VAD ÄR SYFTET MED SYSTEMDOKUMENTATION?	35
6.3 VAD ÄR KVALITÉ PÅ SYSTEMDOKUMENTATION?	36
6.3.1 <i>Fullständighet</i>	36
6.3.2 <i>Förvaltningsvänlighet / anpassningsbarhet</i>	36
6.3.3 <i>Användarvänlighet</i>	37
6.4 ANALYS AV OLIKA INTRESSENTER.....	37
7 SLUTSATSER	38
7.1 NYA FRÅGOR	38
8 KÄLLOR	39
9 BILAGOR	39
BILAGA 1 MAIN ACTIVITIES	40
BILAGA 2 MASTER ACTIVITY PLAN.....	41
BILAGA 3 DETAILED ACTIVITY PLAN	42
BILAGA 4 PROBLEMDescription.....	43
BILAGA 5 PRODUCT DEVELOPMENT.....	44
BILAGA 6A DATAMODELLEN, LINKS OCH APPLICATION AREA	46
BILAGA 6B DATAMODELLEN, MODEL SPECIFICATION OCH KITS.....	47
BILAGA 7 ORDLISTA	48

1 Inledning

Uppsatsen kommer att behandla de problem som uppstår vid framtagandet av dokumentationen. Vilken information behöver systemutvecklarna, vilka metoder och verktyg finns och används idag? Hur skall dokumentationen se ut för att den ska kunna fungera som ett kontrakt mellan SKF Dataservice AB och kunden?

På ett stort företag, som SKF, finns det standardformat och regler för hur dokumentation skall tas fram och hur den skall utformas. Vad man måste tänka på när projekten ofta sträcker sig över flera länder är att det kan vara svårt att enas och komma överens om vad som ska gälla. Som systemutvecklare får du bud och krav från alla möjliga håll utan att veta om det är bestämt att det är just så det ska vara. En medvetenhet om att människor uttrycker sig och arbetar på olika sätt måste finnas. Det är inte alltid så lätt att anpassa sig. Detta märks mer i arbete tillsammans med flera olika länder, det märks tydligare att människor arbetar på olika sätt och det gör att det kan bli en sorts kulturkrock när de ska fatta beslut tillsammans. Hur ska man samla in och analysera kundkraven för att kunna skriva en Reference Guide¹? Det är kundkraven jag kommer att titta på i denna utredning. Hur man får in dem och hur de dokumenteras.

¹ Se förklaring i ordlistan

1. Inledning

1.1 Bakgrund

En hel del anledningar ligger bakom startandet av det här projektet. Idag använder man bl a ett AS/400 system som heter Croesus som är ett korsreferens-system mellan konkurrerande produkter och SKF produkter. I systemet finns redan sökfunktioner för att identifiera olika produkter. Man kan även få reda på vilka produkter som passar i ett visst fordon. Det finns två sorter användare. Dels uppdaterarna som består av produktansvariga i Industrial Division och Vehicle Parts Organisation, VPO, dels de användarna som refereras till som Retrieveanvändarna, de som endast kan se informationen och göra utsökningar vilka består av SKF:s försäljningsbolag och distributörer.

Ändringar i ett befintligt system måste komma in som ett "Request", dvs en förfrågan från användarna. Hur dessa förfrågningar ska hanteras finns dokumenterat i SKF:s kvalitetsmanualer. Vehicle Parts Organisation, vår kund i det här fallet, skickade ett antal CRD's till oss. En CRD, Change Request Description, är en beskrivning av någonting de vill förbättra med det existerande systemet. Dessa CRD's bedöms och prioriteras av en URG, User Reference Group, vilkas medlemmar representerar användarna. När de krav som kommit in analyserats beslöt URG att det var för mycket arbete som behövdes göras ifall man skulle ändra det befintliga systemet. Istället beslöt de att ett nytt projekt, Compass, skulle startas upp, vilket skulle resultera i ett helt nytt system som även hade som syfte att ersätta det gamla Croesus.

Idag är möjligheterna stora och utvecklingen går snabbt framåt, vilket även användarkraven gör. Kraven från användarna i det är projektet tas upp här nedan.

VPO vill ha ett produktutvecklings-system som ska vara gemensamt för Göteborg, Singapore, USA och Frankrike. Man vill även ändra strukturen på informationen i databasen, man vill ha en global databas så att man kan dela information och slipper dubbelarbete. SKF:s säljare och distributörer ska få tillgång till all information samt man vill kunna skriva ut kataloger från ett enda system. Idag har man flera system där man lagrar information för att kunna skriva kataloger. VPO vill också genomgående även få en bättre kvalitet på sin information.

Användarna vill frångå det gamla, AS400 med svartvita skärmar och vill istället ha ett grafiskt interface, ett nytt utseende på systemet. De vill ha ett PC-interface med all den funktionalitet som Windows har. Samtidigt vill man ta vara på den funktionalitet som finns på AS400 och med hjälp av ny teknik göra arbetsprocessen smidigare.

1. Inledning

1.2 Problemområde

Fördelarna med AS/400 är många, däribland säkerhet, stabilitet och framför allt driftsäkerhet. För närvarande undersöker IT-avdelningen ifall det är möjligt att fortsätta att ha databasen på AS/400 och integrera den med moderna utvecklingsverktyg. Det finns mycket affärslogik i de existerande systemen som man vill utnyttja men istället arbeta mot ett PC-interface med användarvänlighet.

Idag utgår SKF:s metodik utifrån att man skall utveckla system för AS/400 eller stordatormiljö, men det finns inga riktlinjer eller standards för hur dokumentationen då ska gå till vid integration med moderna utvecklingsverktyg.

Jag har valt att skriva om detta på grund av att SKF ansåg att det behövdes en undersökning inom detta område. Det är viktigt för SKF att "hänga med" i den tekniska utvecklingen. Det finns idag inga riktlinjer eller standards för hur man ska gå till väga med dokumentationen. Jag tycker även att det ligger i tiden och det är ett intressant problemområde. Många av dagens, även internationellt sett, stora företag tampas med detta problem idag. Min praktik på SKF har gjort att jag kommit in på just detta område och insett vilket problem det är och det har väckt mitt intresse. Resultatet av den här undersökningen kommer att ligga till grund för utvecklingen av SKF:s kvalitetsprocesser och arbetsmetodik.

1.3 Problemformulering

Hur dokumenteras kundkraven? I det befintliga systemet fanns standards för dokumentationen, hur skall man göra nu, istället?

Räcker den metodik som finns för att även hantera integration med PC-interface så att man kan dokumentera på ett bra sätt, eller behövs det arbetas fram mer, ny metodik för sådana här fall?

Det är främst Definitionsfasen som jag kommer att inrikta mig på eftersom det är i den som den mesta dokumentation sker.

1. Inledning

1.4 Syfte

Syftet med mitt arbete är att skapa förståelse för hur man inom SKF samlar in och analyserar kundkraven för att kunna vidareutveckla ett befintligt system med ny teknik. Materialet /resultatet kommer att användas för att skapa en Reference Guide som ska ligga till grund för det nya systemet "Compass". Jag vill skapa förståelse för hur man arbetar för att ta fram en Reference Guide och vad som gjordes på vägen dit, vilka svårigheter som finns i att samla in och dokumentera alla kundkraven.

1.5 Avgränsningar

Jag kommer inte att undersöka hur man löser detta ute på andra företag utan endast på SKF och i projektet Compass.

Jag kommer inte heller att gå in på den tekniska biten, hur den tekniska lösningen bör se ut eller vilka utvecklingsverktyg som finns, inte heller att utvärdera något av dessa.

Den fas jag kommer att inrikta mig på är Definitionsfasen och subprocessen Product Development Update.

1.6 Metod

Jag kommer att göra en *fallstudie* av ett projekt som heter COMPASS. Detta projekt skall resultera i ett nytt system, med utgångspunkt från det gamla CROESUS. Jag kommer att jämföra det jag får fram genom mitt arbete med det som Patricia Morales i en undersökning² kommit fram till är systemdokumentationskvalité. Jag kommer också att jämföra det sätt som vi arbetar efter med det som rekommenderas i ISDM. ISDM heter den metodik som de använder på SKF. Mer om den finns beskrivet längre fram.

Jag vill även tillägga att koncernspråket på SKF är engelska, ISDM är alltså skriven på detta språk, vilket gör att många begrepp och uttryck inte har någon bra översättning. Därför har jag ibland valt att inte översätta vissa ord, begrepp och uttryck utan låter de vara kvar på sitt ursprungliga språk.

1.6.1 Varför valde jag fallstudie?

Jag valde att göra en fallstudie kanske främst för att jag, tack vare praktiken, redan var någorlunda insatt i projektet COMPASS. Men också på grund av att det tedde sig självklart: Jag ska ta reda på hur det går till på SKF, följa och studera ett specifikt projekt, vad kan då vara bättre än att göra en studie?

² Examensarbete av Patricia Morales vid Institutionen för Informatik

2 Beskrivning av kunden, projektet och metodiken

Jag kommer här först presentera vem kunden är och vad kunden gör, jag kommer även att kort beskriva projektet som ska resultera i den nya systemet. Därefter kommer jag att gå igenom och presentera den dokumentationsmetodik som finns idag, som endast gäller utveckling av system på AS/400. Vilka är målen och syftena? Hur ser ett projekts gång ut? Vilka olika faser finns? Detta gör jag för att ge läsaren en förståelse för resten av uppsatsen.

Vidare kommer jag ta reda på kundkraven för det nya systemet och se efter vad/vilken funktionalitet som finns i det existerande systemet som kan användas även i det nya. Jag kommer även att jämföra vårt arbetssätt, hur man utvecklar i projektet Compass, hur går dokumentation av kundkraven till, med det som finns beskrivet i ISDM. Därefter kommer jag att sammanställa alltihop och dra slutsatser.

2.1 VPO

Vehicle Parts Organisation är vår kund. VPO är en del av SKF:s Automotive Divisions. Deras produktutveckling är att ta fram och plocka ihop ”kits” som är deras produkt som de säljer till eftermarknaden. Ett kit består av ett lager plus låsande eller tätande detaljer. VPO har tre stycken produktionsutvecklingscentra, ett i Sverige, ett i Singapore och ett i Frankrike. Det är de som är produktutvecklarna och uppdaterarna i det gamla Croesus. Det är även de som publicerar katalogerna till eftermarknaden som ges ut till alla SKF:s säljbolag och distributörer.

2.2 COMPASS

Compass är namnet på det projekt som ska resultera i ett nytt system som kommer att bygga på det gamla Croesus. Genom att använda den affärslogik som finns i Croesus, utnyttja funktionaliteten som AS/400 har samt ge det hela ett PC-interface får man ett användarvänligt system som samtidigt är driftsäkert och stabilt. Kraven från användarna, som även är målen, är att ha ett gemensamt produktutvecklings-system för Göteborg, Singapore, USA och Frankrike. Strukturen på informationen ska effektiviseras, databasen ska vara global så att man kan dela information och slipper dubbelarbete. SKF:s säljare och distributörer kommer att få tillgång till all information och slutligen, kanske den största biten, man vill kunna skriva ut kataloger från *ett* system istället för flera som man är tvungen att göra idag. Samt genomgående vill man få en bättre kvalitet på informationen.

Sammanfattningsvis, mål och syften med Compass:

- * Framställa kataloger.
- * Eliminera dubbelarbete, informationen ska endast behöva läggas in EN gång.
- * Effektivisera och stödja processen.
- * Sprida och dela information globalt.
- * Resultatet ska kunna användas av alla säljare runt om i världen.
- * Täcka in mycket information, vilket gör SKF konkurrenskraftigt.
- * Kvalitetssäkring.

2.3 Hur tas dokumentationen fram idag?

Den metod som de idag använder på SKF för att ta fram och presentera dokumentation heter ISDM. "SKF Information Systems Development Manual". Det är en metod som SKF själva har arbetat fram. Den första versionen kom 1978 och den senaste är ifrån februari 1996. Den gäller endast för utveckling av system på AS/400 eller i stordatormiljö. För PC-interface eller integration med PC finns idag inga riktlinjer eller standards. Utvecklarna gör så som de anser vara bäst från fall till fall.

2.3.1 Mål och syften med ISDM

Manualen är tänkt att användas som en guide när man har en idé eller ett problem som initierar till att utveckla ett system. Den skall följaktligen kunna användas från att man har ett problem till dess att en fullständig applikation är utvecklad och installerad. Om man följer den här manualen är det meningen att applikationen kommer att uppfylla följande:

- * utvecklad enligt SKF:s kvalitetsstandards
- * effektivt utvecklad och dokumenterad
- * installerad och lätt att underhålla
- * accepterad och utnyttjas effektivt av användarna

ISDM behövs som målsättning för att informationssystemet ska få hög kvalitet och för att försäkra sig om att det stödjer SKF:s regler och standards.

2.3.2 Fem användningsområden

- ❖ Utbildning Exempelvis som introduktion av nyanställda i SKF:s sätt att arbeta.
- ❖ Planering Som guide igenom ett komplett systemutvecklingsprojekt eller en fas av ett projekt.
- ❖ "Körning" Standardiserade checklistor och liknande kan användas under hela projektet. Varje avdelning inom SKF kan ta ut för dem relevanta delar.
- ❖ Som Databas I manualen finns riktlinjer och råd, baserade på tidigare erfarenheter från både SKF:s egna avdelningar och externa källor.
- ❖ Som Referens Kan refereras till under *reviews*³.

³ se ordlista för förklaring

2.3.3 Allmänna användningsområden

SKF använder datoriserade informationssystem inom många områden. Exempelvis försäljning, marknadsföring, ekonomi, materialflöde och tillverkning/produktion. Manualen är främst framtagen med dessa i åtanke men är användbar även inom andra områden.

Manualen kan även komma till nytta vid systemunderhåll där den större delen kan liknas vid arbetet som sker vid ny utveckling. Här kan man hitta riktlinjer att gå efter.

2.3.4 Vilka berörs av ISDM?

ISDM vänder sig till all personal som är involverad i utvecklingen och underhållet av datorbaserade informationssystem. Dessa är framför allt:

- Projektets ledare samt medlemmar.
- Analytiker, programmerare och liknande.
- De personer som har förfrågat/begärt att ett informationssystem skall utvecklas eller köpas in.
- Styrelsens medlemmar.
- Användarna av systemet.
- Alla andra som är involverade i utvecklingen av systemet.

2.3.5 Projekt där ISDM är användbar

ISDM ska användas när det är frågan om att bygga eller *utveckla* ett helt *nytt* system, vid modifiering eller *anpassning av existerande* system samt vid *implementering* av inköpta *standardpaket*. ISDM är framtagen främst för utveckling av AS/400 system eller system i stordatormiljö.

2.3.6 Uppgifter där ISDM är användbar

För de allra flesta uppgifter kan man hitta användbara riktlinjer och råd i manualen exempelvis vid framtagning av kravspecifikationer, och processing rules⁴ samt vid testning.

För de allra största projekten eller uppgifterna, kan ytterligare regler behövas som är specifika för just det projektet, alternativt uppgiften.

⁴ Se förklaring i ordlistan

3 Projektöversikt, de olika faserna enligt ISDM

Jag kommer här nedan kortfattat beskriva de faser som täcker ett projekts livstid. Jag inriktar mig på definitionsfasen men alla faser beskrivs för sammanhangets skull. Hur den dokumentation, som ISDM refererar till, ska se ut regleras av olika manualer som är utarbetade efter SKF:s kvalitetsmanualer. Även vilka tester som skall göras finns beskrivna där.

Project Initiation	Pre-study	Definition	Realization	Implementation	Operation & Maintenance
--------------------	-----------	------------	-------------	----------------	-------------------------

3.1 Project Initiation

Syftet med den här första fasen är att undersöka idéer, möjligheter, problem, krav och så vidare för att se om det är någon idé att starta det här projektet. Om inte så händer inte så mycket mer, om man bedömer att genomförbarheten och lönsamheten är tillräckliga går man vidare till nästa fas. Man gör även en riskutvärdering för att se på riskerna, är de för stora kanske det inte är värt att starta projektet.

Resultatet av undersökningen är ett förslag, eller riktlinjer, för projektet, projekt direktiv. Det gör man genom att använda sig av olika analyser: Problem-, Informations- och Beslutsanalyser. Man bör identifiera och utvärdera den troliga lönsamhetens omfattning och fördelar, storleken och kostnaderna av projektsatsningen.

Sedan rekommenderas man även ta hänsyn till vissa punkter.

- Huvudsakliga mål och omfattning
- Enligt verksamhetens och I.S mål och strategier?

För att kunna göra en korrekt bedömning av resurs- och tidsbehov, samt de tillkommande kostnaderna, bör man även göra upp en del alternativa tillvägagångssätt. De ska inkludera alternativa systemlösningar, alternativ omfattning, alternativ mjukvara och hårdvara, alternativa programspråk samt alternativa personalmöjligheter. För dessa alternativ ska man göra beräkningar över min och max samt riskbedömning.

Det beslut som fattas på det här tidiga stadiet är om man ska starta projektet eller inte.

3.2 Pre-study phase

Syftet med den här, den egentliga första fasen i projektet, är att komma med beslutsunderlag för om man ska gå vidare till nästa fas. Underlaget för det beslutet består av kostnads- och tidsbegränsningar för de olika fortsättningsalternativen.

Man skall även ta fram vad systemet ska täcka samt vad systemet ska göra. Och för att få rätt omfattning och ambitionsnivå är det nödvändigt att göra en avgränsning. Dvs vad systemet INTE ska täcka eller göra. För att kunna få en uppfattning av kostnader och fördelar bör man även göra en ungefärlig plan över HUR systemet kommer att arbeta, dess huvudsakliga funktioner och även den tekniska implementationen.

En del möjliga sätt av hur implementationen ska gå till bör också göras klart. Det kan inkludera en undersökning över vilka obligatoriska funktioner och data som måste vara med i den inledande implementeringen. Här är det viktigt att koncentrera sig på de inledande aktiviteterna, så som att definiera mål, omfattning och krav. Man bör försöka undvika att gå in på för mycket detaljer rörande funktion och teknik.

Resultatet blir en rapport som beskriver vilka för och nackdelar som finns för de olika fortsättningsalternativen, av vilka ett kan vara att avbryta projektet. Den här rapporten kallas lämpligen "Pre-study Report".

Vanliga fallgropar kan vara att man gör för detaljerade analyser eller att man drar slutsatser för snabbt.

Beslut tas, baserat på Pre-study Report där för och nackdelar finns för de alternativa lösningarna, om man skall gå vidare med en Definitionsfas eller inte, samt med vilken budget och tidsram. En del systemdokumentation kommer ur den här fasen och eventuellt en ungefärlig och preliminär användarmanual.

I riktigt små projekt går Pre-study och Definitionsfasen ihop och tar väldigt lite tid i förhållande till hela projektets gång.

3.3 Definition Phase

Den här fasen är mycket lik den tidigare, enda skillnaden är egentligen att allt sker mer detaljerat.

Syftet här är att göra en mer detaljerad och omfattande analys än i den tidigare. Alltså även här komma med beslutsunderlag för fortsättningsalternativ. I slutet av fasen ska det vara möjligt att beräkna den totala kostnaden, eller åtminstone arbetskraftsbehovet av det kvarstående arbetet. Definitionsfasen bör normalt uppta 15-20% av projektets resursbehov.

Även den här fasen skall ta fram vad systemet ska täcka samt göra. För att kunna få en uppfattning av kostnader och fördelar bör man även göra en ungefärlig plan över HUR systemet kommer att arbeta, dess huvudsakliga funktioner och även den tekniska implementationen.

En del alternativ till hur implementationen kan gå till bör också göras klart. Det kan inkludera en undersökning över vilka obligatoriska funktioner och data som måste vara med i den inledande implementeringen.

Det är viktigt att utvärdera alla alternativa lösningar. Beroende på det valda alternativet för huvudsystemet får man olika sätt att installera systemet, vilket innebär olika sätt att ladda in filer, temporärt interface osv. Och det i sin tur betyder olika tids- och resurskrav.

Resultatet blir en rapport som detaljerat beskriver vilka för och nackdelarna är med de alternativa lösningarna. Den här rapporten kallas "System Definition Report". Skillnaden gentemot Pre-study är att man ska föra beslutssituationen ett steg framåt, att klart visa de alternativa riktningarna både vad gäller stora och små funktioner.

Ger man klartecken att gå vidare till nästa fas, System Realization, är det väldigt sällan som projekt stoppas. Vid det här laget ska man ha tänkt på allt och om inget oförutsett inträffar, vet man att det blir ett lönsamt projekt.

För att identifiera datoriserade och manuella operationer kan flödesdiagram uppföras. Oftast ändras flödesdiagrammen under realisationen. Dessa diagram kan fungera som underlag vid resursberäkningsarbetet.

Man bör undvika att dra för snabba slutsatser. Ett datoriserat system är inte alltid lösningen på problemet. Rekommenderat är att lägga ner mycket tid på att hitta många olika lösningar som uppfyller målen. Tidsplanering och implementeringen bör gås igenom noga och möjliga sätt att genomföra implementeringen skildras i stora drag.

Med System Definition Report som underlag fattar man beslut om fortsättningsalternativ. En del systemdokumentation kommer ur den här fasen och även en preliminär användarmanual.

3.4 Realization Phase

Med tidsramen och projektdirektiven gällande både systemet och projektet, som beslutats i Definitionsfasen, ska det kompletta systemet:

- designas in i detalj
- konstrueras
- testas till den grad att det är klart för implementering

Man ska kort och gott skapa systemet.

I den här fasen bestäms exakt hur systemet ska konstrueras, själva realisationen av systemet, dvs skriva programmen, utarbeta de manuella operationerna, databaser, filer och "output". Man testar både de individuella systemkomponenterna och systemet som helhet.

Designarbetet, som börjar med resultatet från Definitionsfasen, berör både generell systemarkitektur och detaljerad design av individuella komponenter. Detta arbetet tillämpas på själva systemet och implementationen. Ifall det rör sig om en "färdig" (inköpt) applikation fokuserar Realisationsfasen på installation och testning av paketet och ser till att dokumentationen är fullständig. Förutom detta kanske man vill modifiera det inköpta paketet. I så fall blir även design, konstruktion och testning av interface nödvändiga för dessa ändringar.

Tidigt i den här fasen skall följande fastställas:

- den bästa implementeringsstrategin
- den bästa testningsstrategin
- vilket samspel som behövs mellan olika subsystem eller delar av systemet, och behovet av tillfälliga interface mellan dem

Resultatet är ett system som är väl testat, av hög kvalité och där endast produktionstestning kvarstår. Detsamma gäller för rutiner för filer, databaser, tillfälliga interface och andra komponenter som behövs för själva implementeringen. Övningsmaterial samt användarmanual har tagits fram och man har tränade användare. Implementeringsplanen är vid det här laget detaljerad och välplanerad. Man bör nu även ha en komplett systemdokumentation, som huvudsakligen finns förtecknad i The Data Dictionary⁵.

Innan själva implementeringsaktiviteterna börjar måste man besluta om den slutliga implementeringsstrategin.

⁵ Se förklaring i ordlistan

3.5 Implementation Phase

Här ska man implementera själva systemet och försäkra en effektiv användning. Syftet är att göra den sista systemtestningen i produktionsmiljö och att effektivt implementera systemet. Detta inkluderar vanligtvis en del första användarsupport. En sista test i produktionsmiljö syftar till att upptäcka, och åtgärda eventuella risker för fel, lagringsutrymmesproblem och performance⁶ problem, som kanske kan ha förbisetts tidigare.

För stora system måste implementationen vara noga förberedd och användarna tränas. Omedelbart efter implementeringen bör användarsupport finnas, samt information om resurser och performance samlas in. Samma information samlas in efter en tid och jämförs med den första.

Resultatet är ett välplanerat system med en effektiv användning. Systemdokumentationen och användarmanualen är kompletta och "up-to-date". Erfarenhet att dra lärdom av i framtida liknande projekt ska samlas in och dokumenteras.

När systemet implementerats och körts ska ett beslut tas angående ansvar och resurser för nästa fas, Operation & Maintenance. Systemet ska bli formellt accepterat och projektet upplöses.

3.6 Operation & Maintenance

Syftet med System Operation är att uppfylla verksamhetens krav. Om systemet inte gör det ska det om möjligt ändras eller ersättas av något nytt. Syftet med System Maintenance är att försäkra att systemfunktionerna fortsätter att sträva mot målen som satts upp tidigare.

Användarerfarenheter bör samlas in i en Operational Problem Log och kan summeras i Potential System Modifications för att senare generera i en System Request. Den används för att korrekt kunna utöka systemet. Man skapar implementerings- och ändringsrutiner, man gör sk "afterstudies" och man upprättar en kommunikationskanal med användarna för att få feedback.

Det är omöjligt att vara beredd på alla förändringar men en bra design och utveckling förutser väntade eller troliga ändringar och ser till att strukturen blir lätt att underhålla. Stora ändringar på ett existerande system bör göras som ett projekt.

Man gör en del uppföljningar efter att systemet implementerats och körts ett tag.

I.S Follow-Up:	För att få erfarenhet av I.S-projekt.
System Follow-Up:	För att kontrollera submålen.
Business Follow-Up:	För att se effekten av system i SKF:s verksamhet.
Quality Follow-Up:	För att kontrollera kvalitén av system i produktion.

⁶ Se förklaring i ordlistan

3. Projektöversikt, de olika faserna enligt ISDM

Ett generellt tillvägagångssätt eller närmande bör vara att sträva mot nya, bättre och billigare system istället för att underhålla gamla ihoplappade.

3.7 Sammanfattning Project Phases

Project Initiation

- Sker innan projektet startar
- Genomförbarheten och lönsamheten bedöms, om projektet startas

Resultat: Feasibility Study Report

Pre-Study

- Projektet startas
- Skapar en skiss över systemet och Project Plan för olika alternativ

Resultat: Dokumentation, skissen över systemet
Pre-Study Report

Definition

- Skapar en skiss över systemet och Project Plan för en del alternativ

Resultat: Viss systemdokumentation
Utkast av användarmanual
Defintion Report

Realization

- Specificerar i detalj och skapar systemet
- Testar systemet
- Gör färdigt dokumentationen
- Output

Resultat: Ett väl testat system av hög kvalitet där endast produktionstestning kvarstår.
Inladdningsrutiner för filer och databaser
Övningsmaterial
Komplett användarmanual
Komplett systemdokumentation

Implementation

- Installerar systemet och laddar in filer
- Tränar användarna samt ger support
- Output

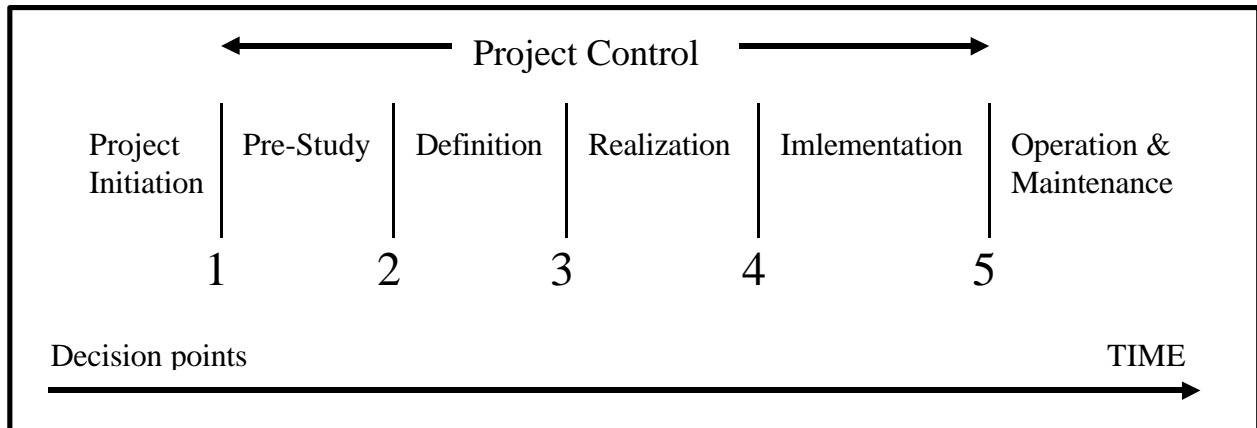
Resultat: Installationsguide
Ett väl implementerat system
Körschema och instruktioner

Operation & Maintenance

- Sker efter det att projektet avslutats
- Sköter om, underhåller, utökar och realiserar fördelaktiga förändringar i systemet

3.7.1 Projektbeslut

Här nedan är en översikt över de beslut som ska fattas och vad man bör tänka på och när.



1. Pre-Study Phase

- Ska man starta projektet eller inte?
- Beslut angående undersökning.

2. Definition Phase

- Grundat på Pre-Study Report, som innehåller för och nackdelar för de alternativa lösningarna, beslutas om man ska gå vidare till Definitionsfasen eller inte. Man bestämmer även i så fall med vilken budget och tidsram.
- I mindre projekt går ibland Pre-Study och Definition ihop till en fas.

3. Realization Phase

- Grundat på Definition Report, som innehåller för och nackdelar för de alternativa lösningarna, beslutas vilka som blir fortsättningsalternativen. Ett utav dem kan vara att avsluta projektet.
- Beslut angående investering.

4. Implementation Phase

- Före själva implementationen, startas arbetet med att få fram en strategi och plan för implementationen.

5. Operation & Maintenance

- När implementationen är klar och systemet är i gång, tas beslut om ansvarsområde och resurser för Operation och Maintenance. Systemet blir formellt accepterat och projektet avslutas.

4 Main Activities

De byggblock som ISDM består av är the Main Activities, huvudaktiviteterna. En sådan är en enda, stor, arbetsuppgift. Vissa av huvudaktiviteterna finns i varje fas i projektet, såsom Plan and Prepare Development Environment. Andra återfinns i flera faser men på olika detaljnivåer, exempelvis Function & Structure.

På grund av att de är ganska många har de olika aktiviteterna tilldelats var sitt ID-nummer (0-99) för att underlätta identifikationen. Se Bilaga 1.

Jag kommer att kortfattat beskriva de aktiviteter som berör våra arbetsuppgifter inom projektet Compass, och som har med dokumentationen att göra (de flesta inom ramen av Definitionsfasen). De är:

- 01 Planning
- 02 Assignment Dialogue
- 03 Follow-Up & Control
- 13 Compile Program Specifications
- 25 Define Objectives and Constraints
- 40 Define & Design System Functions and Structure
- 45 Define Business Functions & Procedures
- 47 Design Screen and Dialogue

4.1 Planning (01)

Det finns två syften med planeringen.

- * dels att göra planer som kommer att ingå i besluts materialet när beslut fattas angående kvalitets- tids- och kostnads målsättning för projektet eller uppgiften
- * och dels att göra planer som talar om hur man ska organisera och schemalägga resurser och arbetskraft, på ett optimistiskt och realistiskt sätt, för att nå upp till den beslutade målsättningen (för kvalitet, tid och kostnader)

Resultatet av planeringen bör vara:

- En lista över grundläggande planeringsförutsättningar, såsom rörande riskfaktor.
- En Master Activity Plan ska skapas. Den är översiktlig över hela projektet.
- Även en Detailed Activity Plan som är på en nivå av enstaka aktiviteter och personer.
- En Resource Plan visar tillgänglighet och uppbokning per tidsperiod och person.
- En Cost Plan visar den tid som man planerar lägga ner per tidsperiod.

En tidsplan ska innehålla milstolpar. Med milstolpar menas ett antal main activities och deras färdigdatum. Utan milstolpar är det svårt att hålla sig i rätt riktning och i rätt hastighet. Varje projektmedlem bör känna till både generella milstolpar och de individuella.

4. Main Activities

I slutet av varje projektfas ska man skapa en generell och komplett projektplan. Den ska fungera som beslutsunderlag när man ska besluta om fortsättningen för projektet. Den bör vara ganska detaljerad för den närmaste framtiden och mer ungefärlig för återstoden.

På samma sätt bör man se över den generella planen i början av varje projektfas. Det är nödvändigt för att kunna reflektera över de beslut som fattats tidigare och för att kunna göra en mer noggrann grund för det fortsatta arbetet.

Det rekommenderas att upprätta och gå i genom Detailed activity plans regelbundet. Det är alltså viktigt att planera. Ofta planeras det för sent och för sällan, om alls. Planering är väldigt sällan slöseri med tiden, som vissa tycker ibland, eftersom det gör klart vilket sätt som är det optimala arbetssättet, vad som kan göras parallellt, vilka aktiviteter som måste startas i tidigare skede osv.

4.2 Assignment Dialogue (02)

Syftet är här att få en specificering av åtminstone den nästkommande projektfasen eller uppgiften. Den här specifikationen, Project Directive eller Contract, ska projektledaren, ägaren av projektet och the Steering Committee komma överens om.

På den här aktiviteten bör man lägga ner en hel del tid precis innan eller i början av varje projektfas. En del planering kan med fördel göras parallellt. Annars kan det vara svårt att sätta upp realistiska mål för resurser, kostnader och tidsplaner.

Lista upp alla händelser, de saker som är viktiga för att lyckas med projektet/uppgiften. Det innefattar alla saker som bör ingå i projektdirektivet (project directive), de är huvudsakligen:

- * Business and Product Quality objectives (verksamhetens mål)
- * En lista över alla krav
- * Det förväntade resultatet; beskrivning av omfattning och gränssnitt
- * Organisation och ansvarsområde
- * Tidsplan med milstolpar
- * Risk/osäkerhetsfaktorer (även hur de ska hanteras)
- * Budgetens ram
- * Resursplan och förbindelser
- * Change Request Handling procedure (hur man ska hantera begärda ändringar)

4.3 Follow-Up and Control (03)

Uppföljningen innefattar tre aspekter; Resultat (kvalité och kvantitet), Resursförbrukning och Kostnad. Det är också bra att följa upp *hur* aktiviteterna har utförts för att nå resultatet.

Syftet med att följa upp resultatet, vilket är samma som att kontrollera ifall en aktivitet är klar eller hur mycket arbete som i så fall återstår, är att göra det möjligt med korrigeringar eller förbättringar. Detta för att kunna uppfylla projektets, systemets och verksamhetens mål.

Även de resurser som används och kostnader som uppstått ska följas upp. Det bör göras för att kunna bestämma ifall korrigeringar behövs. Ibland kan man behöva konsultera The Steering Committee⁷ ifall det rör sig om stora förändringar som att minska omfattningen för att hålla sig inom budgeten eller om man istället ska öka budgeten.

Koncentrationen bör ligga på det som är klart. Det är bra att ha en lista över de systemkomponenter som är 100% klara. Man har ingen större nytta av att veta vad som nästan är klart, kanske till 80% eller 90%, eftersom det fortfarande är osäkert.

Uppföljningarna ska göras regelbundet. Lämpligtvis veckovis, på den mer detaljerade nivån och månadsvis på den generella nivån.

4.4 Compile Program Specification (13)

Syftet är att samla in all grundläggande information som programmeraren behöver för att kunna göra programmeringen. Det insamlade materialet används även för att göra detaljerade regler för de tekniska lösningarna.

Resultatet ska vara en bunt papper som är uttagna från systemdesign dokumentationen. En del information kan utelämnas ifall systemdesignern och programmeraren är en och samma person. Om systemet inte är särskilt stort och programmeraren har tillgång till all systemdokumentation, behöver inte materialet fysiskt tas ur dokumentationen. Detta resultatet har ett namn, det är en Program Specification.

För att den här aktiviteten ska uppfyllas helt följer en lista nedan som fungerar som en checklista över vilket material som ska finnas tillgängligt och från vilken man väljer ut de punkter som passar för just det här programmet.

- Referenser till programdesign och kodningsstandards
- Skärmdialog

⁷ Se förklaring i ordlistan

4. Main Activities

- Skärmlayout
 - fält beskrivningar
 - input fält (valideringar, obligatoriska eller inte)
 - felmeddelanden
 - output fält
 - menyer
 - osv
- Processing Rules
 - felhantering
 - kopplingar med input/output
- Generella felmeddelanden
- Säkerhetsregler
- Updateringsregler
- Funktionstangenter
 - ENTER = uppdatera databasen
 - PFy = hjälp
- Vanliga moduler
- Information från utvecklaren (designern)
- Input
 - parametercards
 - validering av input
 - databaser (segment som berörs)
 - andra filer
 - layout för all input data
 - osv
- Output
 - databaser (segment som berörs)
 - andra filer
- List outputs
 - layout med fältbeskrivningar
 - fördelning och summering
 - antal kopior

Det är inte brukligt att underhålla programspecifikationer efter det att systemet implementerats hos SKF. Men vissa delar behöver ändå underhållas, de ska samlas i en I.S Oriented Processing Rule.

4.5 Define Objectives and Constraints (25)

Det huvudsakliga syftet är att fastställa vilka som är I.S och verksamhetens mål och restriktioner för systemet. Det görs eftersom de kommer att fungera som riktlinjer för systemutvecklingsarbetet.

Resultatet dokumenteras i Objectives and Constraints som även kan inkludera sätt att kontrollera att målen uppfylls.

Verksamhetens mål och restriktioner, som syftar på marknadsföring och tillverkning osv, bestäms av användare medans I.S mål och restriktioner normalt bestäms av I.S utvecklingsavdelning tillsammans med användare.

I.S mål kan refereras till de olika systemegenskaperna, såsom information, procedurflöde, processing rules, resurser och utförande. Mål och restriktioner måste vara relevanta och meningsfulla och styra arbetsinsatserna till de områden där de verkligen behövs. Den generella regeln är att "du kan lösa 80% av uppgiften med 20% av arbetsinsatsen".

Målen effekter bör, så långt som möjligt, uttryckas med procent, siffror, kvantiteter, datum osv.

De allra första målen bör inte räknas som klara, de ändras oftast på vägen beroende på vad som är realistiskt / lönsamt att uppnå. Börja inte med att komma med en lösning. Det kan sluta med att man löser fel problem. Det är också oerhört viktigt att alla inblandade vet vilka målen är och deras prioriteringar för systemet (/uppgiften). Målen bör brytas ner och ordnas hierarkiskt eftersom det gör målen lättare att förstå för samtliga inblandade.

Hur ska man då gå till väga? Man bör göra en problemanalys för att försäkra sig om att målen verkligen löser problemen och inte bara symptomen. Lösningen måste inte vara ett datorsystem, vad som egentligen behövs är kanske en omorganisering eller personalutveckling.

Hur kontrollerar man ifall målen har uppfyllts? För att kunna mäta resultatet av projektet och systemet bör lämpliga sätt definieras och uppföljningar tillåtas redan på ett tidigt stadium. Och helst ska även *hur* prestationen ska mätas definieras. Mål som är abstrakta, som "ökad kundservice" kan oftast delas in i delmål för att då kunna mätas och uppföljas.

4.6 System Functions and Structure (40)

Syftet med den här aktiviteten är att bryta ner systemet till mindre delar som gör det lättare att förstå. Syftet är också att identifiera samspelet mellan de olika delarna. På så sätt får man en effektiv systemstruktur. Det är i den här aktiviteten som man specificerar vilka funktioner man behöver för att uppfylla målsättningen och kraven. Även dessa funktioner bryts ner till en preliminär program och manuell operationsnivå. Ett utkast över vilka huvudfunktionerna för programmet är och samspelet mellan dessa ska göras.

Den resulterande strukturen dokumenteras i Structure Chart. För att den ska vara överskådlig bör den ej visa fler än tre funktionsnivåer på en sida. Vanliga funktioner, även kallade moduler framöver, som man identifierat kan listas i en List of Common Modules. Som komplettering till Structure Chart kan även en Functional Description upprättas.

Vilka är då fördelarna med den här struktureringen?

- * det blir möjligt att planera och utveckla de definierade delarna var och en för sig
- * dokumentationen blir uppstrukturerad och lätt att underhålla
- * framtida ändringar blir lätta att göra
- * själva systemet blir pålitligt och lätthanterligt

När man har beslutat om systemprocedurer ska begreppen delsystem och operationer användas framöver istället för funktioner. Vilka operationer (funktioner) som ska skötas manuellt och automatiskt bör bestämmas tidigt för tids- och kostnadsberäkningar.

Ifall de olika delarna av systemet kommer att utvecklas separat bör övervägande tas gällande funktionalitet, sammanhang, interface, effektivitet osv redan i Definitionsfasen. Gränserna för de komponenter som ska utvecklas separat bör vara klargjorda innan utvecklingen delas upp i delprojekt.

I början av aktiviteten rekommenderas att bryta ner det komplexa systemet i mer hanterbara delar. Krav och resultat kan därefter delas upp på de olika delsystemen. Att fortsätta den funktionella nedbrytningen till lägre nivåer är ett bra sätt att analysera och resultatet blir en bra och pålitlig systemstruktur. Dataflödet mellan funktionerna (delsystemen) behövs ritas ut för att därefter kunna definiera procedurerna.

Struktureringen i de tidiga projektfaserna bör ske tillsammans med användarna genom att bryta ner systemet i logiska funktioner (alltså inte dator eller lösningsorienterade funktioner). I de senare faserna bör man istället komma fram till de fysiska delarna enligt den funktionsstrukturen.

Moduler eller subrutiner som är vanliga ska identifieras och formges på ett tidigt stadium, detta för att använda dem så effektivt som möjligt i program och manuella operationer. Det kan vara både användarfunktioner eller tekniska:

4. Main Activities

- Moduler bestående av vanliga manuella operationer, exempelvis generella skärminstruktioner
- Vanliga delprogram, exempelvis table lookups
- Självständiga serviceprogram som förekommer i flera program
- Delar av återanvändbar kod, makron, copy texts osv

Följande riktlinjer bör följas för att utvärdera resultatet av nedbrytningen:

- Funktioner som på samma nivå bör vara så oberoende som möjligt
- Interfaces mellan funktionerna bör vara väl definierade och klara
- Funktionerna bör vara så självständiga som möjligt

Om systemet eller funktionen är komplex eller om den är svår att bryta ner i lägre nivåer, kan man först definiera det huvudsakliga dataflödet. Exempelvis genom att använda dataflödesdiagram. På de ställen där datan strålar samman eller delar på sig bör funktioner definieras.

IPO diagram (Input - Process - Output) kan användas för att få en överblick samt göra det lättare att läsa och förstå.

Bestäm slutligen vilka moduler som ska vara program eller manuella operationer, online eller batch, distribuerade eller centrala osv. Låt datorn hantera alla rutiner och massarbeten och lämna bedömningar och undantag åt människorna.

4.7 Define Business Functions & Procedures (45)

Syftet med den här aktiviteten är att definiera "Business Procedures" som består av företagets affärsområde och för att försäkra sig om att man får tillräckligt med support från informationssystemet och dess procedurer.

På systemnivå blir resultatet Business Procedure Overview Chart och på procedurnivå Business Procedure Flow Chart. Eventuell stödande information kan skrivas in i Business Procedure Description.

Generellt betyder termen "procedure" beskrivningen av en mängd funktioner, deras arbetsförlopp, timing och andra inbördes förhållanden som behövs för att få definitionsarbetet gjort eller en händelse att inträffa. På en generell och hög nivå av definition innefattar begreppet "business procedures" bland annat saker som Stock Replenishment (lagerpåfyllning) och Sales Forecasting (försäljningsprognoser). För att kunna stödja sådana procedurer behövs en eller flera "system procedures". En system procedure, eller delar av den, kan stödja flera business procedures.

I de flesta fall finns det redan business procedures och det räcker att beskriva dem, ifall inte det också redan är gjort. Beskrivningarna ska kontrolleras emot målen som är satta för utvecklingen och eventuella avvikelser ska tas upp för diskussion och beslut fattas om ändringar bör ske av business proceduren eller/och målen. Om det är så att det är existerande business proceduren som ska modifieras eller att en ny ska skapas, är det viktigt att ha ett starkt samarbete med användarna. Detta för att upprätta

4. Main Activities

acceptabla och effektiva procedurer. Tonvikten måste läggas på att minska administrativa eller maskinella resurser och tid. Till exempel analysera vilka operationer som kan utföras parallellt.

Generellt gäller att göra procedurerna så lätta som möjligt, d v s att underlätta arbetet för den personen som ska använda dessa procedurer. Om procedurerna inte går att utveckla så att de blir effektiva kan det bero på att systemstrukturen inte är tillräcklig och kan behövas ses över.

Business Procedure Flow Charts är den vanligaste metoden för att beskriva procedurer och kan stödjas av förklarande beskrivningar. Business Procedure Flow Diagrams kan användas för att beskriva flödet mellan olika organisationsenheter såsom avdelningar och lagerlokaler. Network Diagrams kan användas för att hitta den kritiska (svaga) vägen genom business procedures.

4.8 Design Screen and Dialog (47)

Det huvudsakliga syftet är här att bygga ett användargränssnitt som är enkelt att använda och som sparar tid och arbete för användarna. Detta görs genom att upprätta ett konsekvent sätt att kommunicera mellan användare och dator, som även passar den tänkta användargruppen. Till exempel att terminologin är konsekvent och anpassad för användaren, konsekvent skärmlayout, dialog och metoder.

4.8.1 Design principer

- Användargränssnittet ska bekräfta den konceptuella modellen som användaren kan få av gränssnittet, genom att se till att användaren kan få ut den information/händelse som de förväntar sig.
- Låt användaren kontrollera dialogen. Det innebär att användaren ska kunna utföra vilken handling de vill i vilken ordning som helst, för att kunna utföra sina arbetsuppgifter.

4.8.2 Krav på skärmlayouten

- Den ska vara lätt att läsa och förstå.
- Man ska använda en konsekvent layout på alla skärmar. Den information som visas ska minimeras så att endast det som är nödvändigt för användaren visas.

4.8.3 Krav på dialogen

- olika rutiner för erfarna och oerfarna användare
- så lite monotont arbete som möjligt
- korta svarstider
- lätt att få hjälp av systemet
- feedback för varje handling användaren utför
- möjligheten att ändra sig och ångra den senaste handlingen
- möjligheten att återanvända data från den föregående skärmen
- när kraftfulla handlingar ska utföras, som vid deletering, ska systemet begära en bekräftelse
- gränssnittet ska vara sådant att användaren ser vad du ska göra istället för att behöva komma ihåg det. Där det är möjligt låt användaren välja från listor istället för att komma ihåg de rätta valen.

4.8.4 Skärmdisposition och layout

Skärmen ska bestå av tre delar. Överst systemnamnet, panel titel, datum och tid. Längst ner ska det finnas en functionkey area och meddelande ruta. Utrymmet däremellan är ledigt för användningen av applikationen.

När ett system ska köras på flera olika ställen samtidigt kan det vara användbart att inkludera den gällande fysiska positionen som en del av systemnamnet. Till exempel SWE - Croesus, där SWE står för systemets position och Croesus är namnet på systemet.

De dialogmetoder som rekommenderas att användas är inmatning och urval. Allt det här gäller endast för AS/400system.

4.8.4.1 Inmatning

- Användarna skriver bokstäver eller siffror i ett inmatningsfält.
- Data matas in i fördefinierade fält. När en skärm först visas kan ett inmatningsfält visas med eller utan fördefinierat värde. Markören placeras i det första input-fältet.
- Först när applikationen identifierat alla inmatningsvärdena utförs händelsen som begärts av användaren.

Ett inmatningsfält identifieras genom

- * fältprompt
- * kolumn rubrik eller beskrivande text
- * grupprubrik
- * prompt information

4.8.4.2 Urval

- Användaren väljer från en grupp av relaterade val som presenteras på skärmen.
- Användaren väljer ett eller flera av de val som presenteras genom att skriva in en bokstav i det fält som associeras med det valet (valen).

Det finns tre sorters urval

- * urvalsfält
- * urvalslista
- * actionlista (lista med handlingar)

En annan typ av urval är när användaren skriver in hela kommandot på en gång, istället för att ta en bokstav i taget och låter systemet leta varje gång, vilket tar mycket tid i anspråk. Den möjligheten finns endast som ett komplement till de andra metoderna, den är aldrig den enda.

Något som man bör tänka på vid implementeringen är att det inom ett system, eller inom de system som vanligtvis används tillsammans, är viktigt att både skärmdesign och dialog är konsekventa. Annars kan det lätt bli förvirrande och användaren tappar förtroendet för hela systemet.

5 Tillvägagångssätt

När man arbetar objektorienterat så som vi, i projektet Compass, gjort är det första man bör göra en datamodell. Det första steget var alltså att samla in all information om alla förekomster som ska bli datamodellen. Sedan har vi gått igenom alla objekt och relationer med användarna vilket till slut ska resultera i datamodellen. Som datamodellen ser ut idag, version 1, återfinns på Bilaga 6a och b.

Jag har gått igenom alla filer i det gamla systemet som bygger på AS/400. Detta för att ta reda på vilka olika fält och vilken funktionalitet som fanns med. Det är de som är de främsta kundkraven, vilken information ville de ha med i systemet från början.

Jag har också samlat in information om alla de fält som användarna tror sig behöva i Compass, det nya systemet. Vi har också hjälpt användarna att specificera själva processen, genom prototyping.

Vi gjorde förslag på skärmlayout där man kan mer praktiskt se hur de vill arbeta med systemet och om vi förstått vad de vill ha.

Dokumentationen som uppstått är bland annat: se Bilagor

- * En Master Activity Plan
- * En Detailed Activity Plan
- * Ett Data Dictionary
- * Ett antal grafer och listor
- * En förteckning över Business and Process Objectives (målen)

5.1 Nulägesanalys

För att kunna göra en analys av nuläget har vi, i projektet Compass, filtrerat ut de problem som är mest akuta, och på så sätt inringat problemområdet och detaljstuderat det. Vilka problem finns idag? Och vart kan de härledas? En del problem kommer ur andra problem, andra uppstår på andra system, existerande system osv. Det gäller kortfattat att komma åt kärnan, det vill säga de riktiga problemen och inte symptomen. På så sätt fick vi en klar överblick av de problemen som vi skulle inrikta oss på, en slags avgränsning av problemområdet.

5.1.1 Beskrivning av de inrapporterade problemen

För subprocessen Product Development Update har jag gjort en problembeskrivning. Tanken med en problembeskrivning är att se på problemen från olika vinklar. Vi har använt oss av en blankett "Problemanalys" (se Bilaga 4). Den består av tre delar, en *problemdel*, en *orsaksdel* och en *konsekvensdel*. När man samlat in och analyserat alla problemen kan man ibland upptäcka att det som vissa personer upplever som ett problem kan för någon annan vara en orsak. En konsekvens kan bli ett nytt problem för någon annan. Därför har vi även, där vi funnit det lämpligt, tagit med *problemområde*. De problemområdena är antingen Product Development, PD (Update) eller System. PD är kort förklarar det som användarna vill ha med i det nya systemet som inte finns med i det existerande. System innefattar de förbättringar som man vill göra i det existerande. Jag har valt ut området PD eller Update som det även kallas som ett exempel som kan följas.

För att få en klar och överskådlig överblick över de problem, mål och behov som finns samt vilka åtgärder man bör ta, benade vi upp alltihop i olika listor och grafer. Dessa följer nedan och gäller alltså problemområdet PD..

5.1.1.1 Problemlista

Problemen undersöktes för att upptäcka eventuella beroenden dem emellan. Här nedan är problemlistan och mållistan. De ges en bokstav, P för problem och M för mål, detta för att det ska vara lättare att identifieras i graferna som vi sedan gjorde.

COMPASS	Problemlist	Responsible: A-C.H
Problemanalysis	Version: 1.0	
Date: 1998-05-13		
No	Problem	
P1.	The same kit is developed twice, within or in different departments.	
P2.	No messaging and reporting facility.	
P3.	Don't know the kits contents.	
P4.	Data is stored in several different systems.	
P5.	No Product Development system. Info is not used correctly, for example Gothenburg can't use the info that Singapore has.	
P6.	The Modelspecification is not valid.	
P7.	The Applicationlist must be respecified.	
P8.	The users don't like the system.	
P9.	Too many footnotes.	
P10.	Seals crossreference doesn't exist in the system.	
P11.	Bad quality of information.	
P12.	No common workprocedures.	

5. Tillvägagångssätt

5.1.1.2 Mållista

Slutligen kom vi upp med ett förslag till målanalys, Objectiveanalysis. Den listan följer här och ser ungefär likadan ut.

COMPASS	Objectivelist	Responsible:	A-C.H
Objectiveanalysis	Version:	1.0	
Date:	1998-05-13		
No	Objectives		
M1.	Share information between PD-people.		
M2.	Common system for all VSM Product Development departments.		
M3.	Common Automotive Database containing global information.		
M4.	Include overall common process to create a new product range.		
M5.	Increase product development efficiency.		
M6.	Enter information only once.		
M7.	Several possibilities to view information.		
M8.	Better diffusion (easier, quicker) of info to all departments concerned.		
M9.	A daily working support for all concerned departments.		
M10.	Messaging and reporting facility.		
M11.	Userfriendly system.		
M12.	Links to other systems. Internal system interaction Telecom Customer Links		
M13.	Platform independent		
M14.	Updating possibilities for other divisions.		
M16.	Better than existing system.		
M17.	Secure system.		
M18.	Clear responsibility for updaters.		
M19.	Different levels of security.		

5. Tillvägagångssätt

5.1.1.3 Förändringsbehov

Det som sedan återstår är att koppla problemen till de olika målen och att föreslå behov av förändringar och åtgärder, en behovsanalys och en åtgärdslista. Här benar man även upp vilka problem som kan lösas genom datorisering och vilka problem som är business processrelaterade.

Så här kom vår behovsanalys att se ut. Även här identifieras de olika behoven av en siffra (R står för Request). Exempelvis R1, ny struktur för databasen, innefattar problem nr 6, 7, 8, 9, 5 och 11, de i sin tur uppfyller mål 2, 3, 7, 17, 5, 6, 8 och 16. Eftersom R1 löser rätt många problem och uppfyller många mål är det något att koncentrera sig på.

COMPASS	ChangeRequest	Responsible:	A-C.H
Requirementanalysis	Version:	1.0	
Date	98-05-13		
<u>No</u>	<u>ChangeRequest</u>	<u>Problem</u>	<u>Objectives</u>
R1.	New database structure.	P6,P7,P8,P9,P5,P11	M2,M3,M7,M17 M5,M6,M8,M16
R2.	Windowsinterface	P8	M7,M16
R3.	Productdevelopment update module.	P1,P2,P3,P4,P5,P11	M8,M9,M14, M16
R4.	Links to MRPSystems, Prodmast.	P4,P11	M12,M16
R5.	Batch Update program (one-off)		
R6.	Messaging and reporting facility.	P2,P11	M10,M16
R7.	Securitylevels	P12,P11	M18,M19,M17
R8.	History log	P1,P11	M1

5. Tillvägagångssätt

5.1.1.4 Åtgärdslista

Därefter gjorde vi en åtgärdslista, Actionlist. Här får vi fram vilka åtgärder som måste göras. Exempelvis A2, gemensam process för utveckling av kits, uppfyller behoven R2 och R3 osv.

COMPASS	Actionlist	Responsible: A-C.H
Actionanalysis	Version: 1.0	
Date	98-05-13	
No	Actions	Involves
A1.	Develop a database with new structure that can be accessed by all PD-people.	R1
A2.	Common process for developing kits.	R2, R3
A3.	Link the databas so that it updates all systems involvoed.	R4
A4.	Batch update of existing information.	R5
A6.	Messaging and reporting facility.	R6
A7.	Securitylevels	R7
A8.	History log	R8

5.2 Objektorienterad Verksamhetsanalys

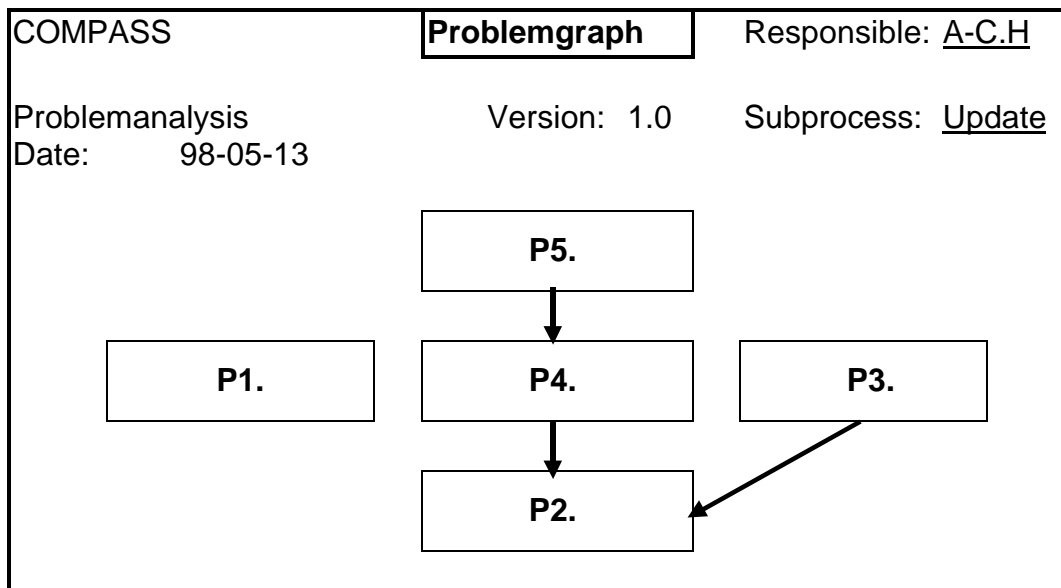
Det som utmärker Objektorienterad verksamhetsanalys är att man på ett tidigt stadium diskuterar problemområdet i objekt-termer. Detta är till fördel både för utredarna, eftersom de tänkta systemanvändarna känner igen objekten, och för systemkonstruktionen, eftersom man får en lätt övergång till en implementering i ett objektorienterat språk eller till relationsdatabaser.

Vi samlade in alla problem som vi fått in av kunden och gjorde en Problemanalys. På så sätt blev det enkelt att anteckna problemen och deras orsaker.

5.3 Grafer

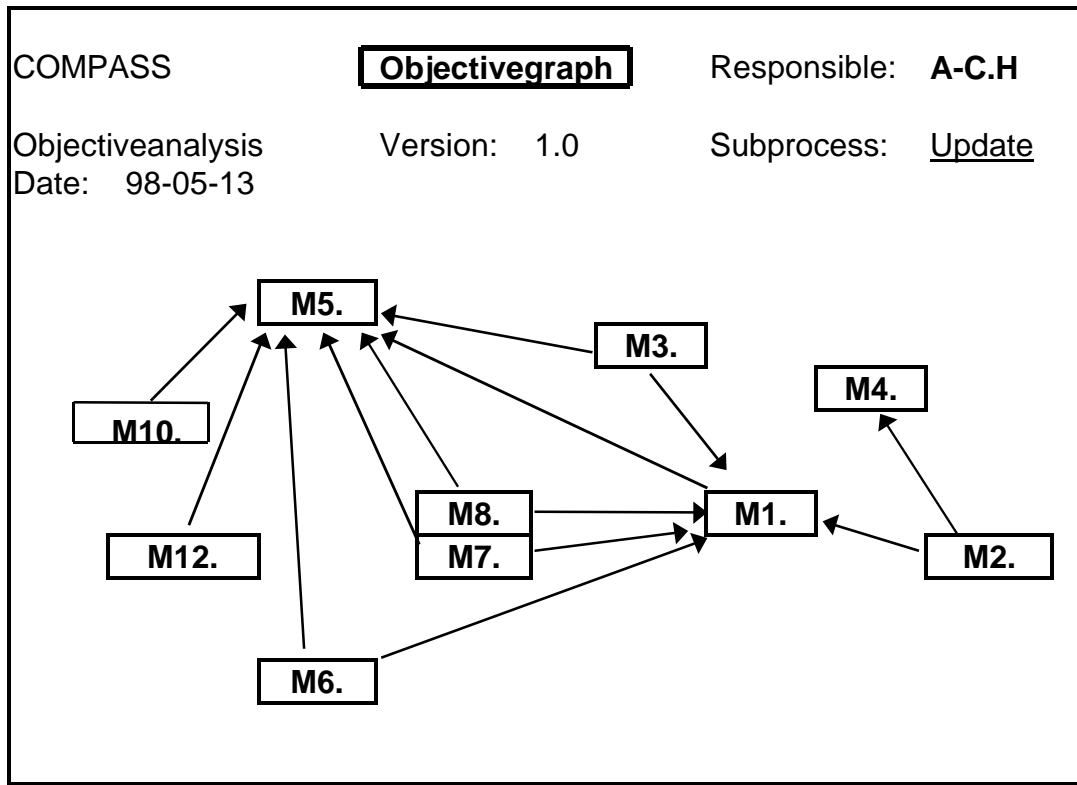
För att få en bra överblick över hur problemen och målen hänger samman, både gentemot varandra och över gränserna, har vi upprättat ett antal grafer. Först ser vi hur problemen är relaterade med varandra och lite längre ner det samma för målen.

5.3.1 Problemgraf



Vi ser att problem nr 5 är relaterat till både nr 4 och nr 2, därför är det P5 man bör koncentrera sig på att lösa eftersom det löser de båda andra också. P1 ligger för sig själv vilket innebär att man måste lösa det för sig och även P3, som då även löser P2.

5.3.2 Målgraf



Här framgår det tydligt att alla målen rör sig runt M5 och M1 som är de viktiga målen (som synes är det lite av detektivarbete).

5.4 Arbetsprocessen

Vi har hjälpt användarna att specificera själva processen genom att göra en prototyp av hur det tänkta systemet skulle kunna fungera och se ut. Vi gjorde upp ett förslag på skärmlayouter dels för att användarna mer praktiskt kan se hur de vill arbeta med systemet, och dels för att kontrollera att vi förstått vad de menar och vill ha ut av systemet. Det var mycket uppskattat.

5.5 Databasdesign

För att komma fram till en datamodell har vi gått igenom alla entiteter och förhållanden dem emellan med användarna. Den är tyvärr inte riktigt färdig i dagens läge, detta beroende på att det kommit till nya krav vilka måste tas hänsyn till. Så som datamodellen ser ut idag, version 1.0 finns som Bilaga 6.

5.6 Jämförelse mellan vårt arbetssätt och ISDM

Jag började med att gå igenom ISDM och listade de dokument och resultat som föreskrevs. Sedan gjorde jag samma sak med det arbete som vi gjort. När jag sedan jämförde de båda sätten kom jag fram till att vi faktiskt har täckt upp de bitar som ISDM föreskriver och fått ungefär samma slutresultat.

Enligt ISDM ska man göra en Master Activity Plan och en Detailed Activity Plan, det har vi gjort. Business och Process Objectivities ska definieras, vilket vi också gjort. En lista över alla krav och vilket det förväntade resultatet är har vi, från användarna, samlat in, analyserat och utvecklat ytterligare. Däremot har vi inte upprättat en Program Specification som det rekommenderas i ISDM. Detta beror på att vi än idag inte vet vilket verktyg vi ska använda eller i vilken miljö vi ska utveckla.

Något som vi gjort, som inte finns beskrivet i ISDM, är prototyping. Vi har genom att göra ett förslag på skärmlayout hjälpt användarna att specificera själva processen, det har gjort att de mer praktiskt kan se hur de vill arbeta.

I ISDM står det att man bör göra en problemanalys, för att försäkra sig om att man verkligen löser problemen och inte symptomen. Tyvärr står det inte hur analysen ska gå till. Därför har vi gjort så som vi tyckt var bäst. Vi har upprättat listor över problem och målen samt åtgärder och grafer som visar hur målen och problemen relaterar till varandra.

Sedan finns det ett par saker som vi inte hunnit bli klara med, tyvärr. Anledningarna till detta tas upp i uppsatsens slutsatser. Det gäller definitionen av Business Procedures och Structure Chart. För den sist nämnda har vi börjat gå igenom entiteter med användarna och den ska till slut leda till en datamodell. Även Business Procedures hjälper vi användarna med men är inte riktigt klar.

Vi har alltså använt ISDM i den mån vi funnit den lämplig men ofta gjort det på vårt eget sätt. Som jag skrev tidigare, det står oftast inte beskrivet hur man ska göra endast vad som ska göras. Det har gjort att vi använt oss av tidigare erfarenheter och gjort saker och ting så som det passade oss bäst. Och i slutändan har vi kommit fram till ungefär samma resultat. Men ISDM behöver helt klart ses över. Den bör uppdateras till dagens förhållanden, de flesta system idag är integrerade med PC, och kanske viktigast av allt, metoderna måste bli klarare och tydligare.

6 Vad är systemdokumentationskvalité?

För att avgöra om systemdokumentationen i ISDM är tillräckligt bra har jag utgått ifrån en tidigare undersökning i ämnet av Patricia Morales⁸, som jag kortfattat beskriver här nedan. Det som hon kom fram till i sin undersökning har jag använt som ett slags facit att kontrollera emot. Vad systemdokumentationskvalité är tycker jag Patricia Morales besvarar klart och tydligt i sin undersökning.

6.1 Inledning

För att ett system ska fungera krävs mänskliga arbetsinsatser och dessa ska stödjas av anvisningar och instruktioner. Systemdokumentationen är den synliga slutprodukten, den beskriver systemet för användaren och måste naturligtvis vara klar innan det är dags att installera systemet.

För att få en bra dokumentation måste användaren förstå innehållet, detta kan uppfyllas ifall användaren själv är med att utformar den.

Systemdokumentation kan delas upp i två kategorier, dokumentationsutveckling och dokumentationsprodukter.

Dokumentationsutveckling beskriver och talar i detalj om vad användarna behöver, deras krav, vad systemet gör och hur systemet gör det.

Dokumentationsprodukter är de produkter som ska hjälpa till i själva användningen av systemet. Det kan vara användarmanual, underhållsmanual eller operatörsmanual.

6.2 Vad är syftet med systemdokumentation?

Systemdokumentationen ska stödja och styra den löpande driften, underlätta för underhåll, samordning och vidareutveckling. Den ska förbättra kommunikationen mellan användarna och interface samt sprida information om systemdokumentationens användning och funktionssätt.

⁸ Examensarbete I "Systemdokumentation-kvalité" av Patricia Morales vid Institutionen för informatik

6. Vad är Systemdokumentationskvalité?

6.3 Vad är kvalitet på systemdokumentation?

En dålig systemdokumentation påverkar hela systemet. ”Om användarna inte kan hitta den information de behöver, på exempelvis dåliga beskrivningar, medför det stora förluster för företaget, det tar tid att utreda orsaken och sedan åtgärda det.” Patricia Morales använder tre begrepp som hon anser måste vara användarkrav på systemdokumentationen; *fullständighet*, *användarvänlighet* och *förvaltningsvänlighet*. Ifall dokumentationen får en hög kvalitet minskar kostnaderna för både drift och underhåll, vilket naturligtvis är det som alltid eftersträvas.

Detta uppnås genom att man analyserar användarnas informationsbehov samtidigt som man tar hänsyn till erfarenheter från tidigare tillfälle gällande strukturering och utformning av dokumentation. Även terminologin är viktig, användarna måste förstå vad som står, hävdar Patricia, man måste undvika fackspråk så mycket som möjligt och skriva så att de som ska använda dokumentationen verkligen förstår. Man kan även utbilda användarna och hjälpa dem genom att sätta dem in i arbetet.

6.3.1 Fullständighet

Specifikationsenligt innebär att systemdokumentationen följer de standardformat och regler som företaget har utarbetat.

Tillförlitlighet är att användarna måste kunna lita på dokumentationen.

Tillgänglighet innebär att dokumentationen alltid ska finnas till hands när den behövs.

Tvingas man gå till exempelvis en annan avdelning kommer den inte att användas, helst ska alla användare ha var sin.

6.3.2 Förvaltningsvänlighet / anpassningsbarhet

Med anpassningsbarhet menar Patricia Morales att systemdokumentation måste vara anpassad till de olika personer som kommer att använda systemet, deras erfarenheter, behov, utbildning osv.

Man uppnår detta genom att välja en modulär uppbyggnad för exempelvis beskrivning av olika funktioner eller ärenden. ”Det gynnar både användare och systemutvecklare, det är alltid lättare att arbeta med en väl avgränsad uppgift som är både funktionell och oberoende.”

6. Vad är Systemdokumentationskvalité?

6.3.3 Användarvänlighet

För att dokumentationen ska räknas som användarvänlig måste två saker vara uppfyllda. Det ska vara lätt att läsa och lätt att hitta.

Lätt att läsa grundas på hur systemdokumentationen utförs. Den måste skrivas på ett klart och koncist sätt och alltid vara anpassad efter vem det är som skall använda dokumentationen.

För att det ska vara lätt att hitta måste den vara lätt att hantera, den ska kunna fungera som ett stöd för användarna och inte tvärt om. Företaget bör analysera och hitta den bästa metoden för att organisera systemdokumentationen så att den passar både företaget och användarna.

6.4 *Analys av olika intressenter*

Att analysera de olika intressenternas krav och behov av systemet är viktigt för att få en välfungerande dokumentation. "Varje användare eller intressent som kommer i kontakt med systemet måste ha den information som de behöver tillgänglig." Två personer som har samma intresse i systemet och samma tekniska bakgrund tillhör samma grupp av intressenter medans om de har olika bakgrund tillhör olika grupper. Det som är svårt vid utformningen och skrivandet av manualer är att den ska stöda både erfarna och nya användare.

Alla manualer måste vara fria från missledande information, som felaktigheter i satskonstruktioner och grammatiska fel eftersom detta får användarna att tappa förtroendet både dokumentationen och systemet. Det går åt en massa tid till att ta reda på vad som gick fel och tid är pengar.

7 Slutsatser

I och med att ISDM inte innehåller några som helst riktlinjer för hur man ska hantera dokumentationen vid utveckling av system med PC-integration har vi, jag och systemanalytikern Ann-Christine Hagberg, arbetat fritt och använt oss av metoder och tidigare erfarenheter, bl a från skoltiden. I det här fallet har det gått bra men så väl går det nog inte alltid. ISDM är bristande på flera punkter, vilket jag tar upp här nedan, och måste uppdateras till den nivå där systemutvecklingen befinner sig idag.

Ett av mina syften med den här uppsatsen var att identifiera svårigheterna vid framtagandet av dokumentation. Därför har jag jämfört ISDM med vad Patricia Morales har kommit fram till är systemdokumentations-kvalité och kommit fram till att ISDM helt klart behöver ses över. Den är svårläst, det är svårt att hitta den information man behöver, de metoder som beskrivs är inte klara och terminologin är inte anpassad för andra än de som är insatta i SKF:s rutiner. Allt det här har bidragit till att vi inte arbetat helt enligt ISDM. Ofta står det endast vad det är som ska göras, sällan står det beskrivet hur man ska gå till väga. Därför skiljer sig vårt arbetssätt från det som ISDM rekommenderar. Vi har valt ut det som passade oss. Med tanke på att SKF använder sig av en hel del konsulter utifrån, som inte är insatta i SKF:s rutiner så borde detta ändras. ISDM skulle även behöva uppdateras, där står mycket som kanske inte längre är så aktuellt. Man behöver moderna utvecklingsmetoder som klarar av dagens krav på snabb och effektiv systemutveckling.

Även ytterligare standards behöver arbetas fram. Det gällande utveckling i andra miljöer än stordator och AS/400, idag är det integration med PC som gäller och för det räcker de standards och riktlinjer man har idag inte på långa vägar.

Jag har även kommit fram till att det tar längre tid än planerat att utveckla system på SKF. De Main Activities som jag gått igenom är de som vi planerat hinna med men så är tyvärr inte fallet. Detta beror främst på att projektet har växt under tiden och ytterligare krav har kommit in. Utöver VPO har andra delar av SKF:s organisation också velat ha med sina krav i det nya systemet. Det gör att systemet måste hantera fler krav än vad som var tänkt från början. Sedan har vi fått en ny projektorganisationen på användarsidan vilket också har tagit ytterligare tid. Från början var vi bara 2-3 personer från IT-sidan som var inblandade, nu är vi 11. Ännu en sak är att man stött på problem vad gäller den tekniska lösningen, som skett parallellt. Allt det här har tillsammans gjort att det tagit betydligt längre tid än vi räknat med.

7.1 Nya frågor

Exempel på nya frågor kan vara vilka standards, som även inkluderar utveckling av system med PC-integration som ska tas fram och hur detta ska gå till.

8 Källor

ISDM - SKF Information Systems Development Manual

”Systemdokumentation-kvalité” Ett examensarbete I av Patricia Morales

9 Bilagor

Bilaga 1	En lista över alla Main Activities
Bilaga 2	Master Activity Plan
Bilaga 3	Detailed Activity Plan
Bilaga 4	Problemdescription
Bilaga 5	Product Development (från kunden, vilka krav de har och vilka problem de har mm)
Bilaga 6a, b	Datamodellen
Bilaga 7	Ordlista

Bilaga 1

Main Activities

- 01 Planning
- 02 Assignment Dialogue
- 03 Follow-Up & Control
- 04 Plan & Prepare Development Environment
- 05 Train Project Personnel
- 06 Organization & Staffing
- 07 Keeping Parties Concerned Informed
- 09 User Training

- 11 User Manual
- 13 Compile Program Specifications
- 21 Determine Requirements on the system
- 23 Fact Gathering and Analysis

- 25 Define Objectives and Constraints
- 29 Long Term Considerations
- 31 Define Scope and External Interdependencies

- 35 System Alternatives & Packages

- 40 Define & Design System Functions and Structure
- 42 Systems Interface Design
- 45 Define Business Functions & Procedures
- 46 Define & Design System Procedures
- 47 Design Screen and Dialogue
- 48 Generalization

- 50 Define Information Sets and Data Contents
- 51 Define Data Elements
- 53 Information Analysis and Data Base Design

- 55 Data Base Initiation, Monitoring and Tuning
- 56 Design Files and Tables
- 57 Initial Data Base Loading
- 58 Reports & Other I/O
- 59 Use of Data Dictionary/ Repository
- 60 Define Processing Rules
- 63 Detail Manual Operations and Procedures
- 64 Specify Controls and Security Requirements
- 66 Detail & Check System Performance
- 69 System/Project & Database Reviews
- 70 Define Requirements on Organization
- 72 Define Requirements on Hardware and Software
- 73 Software Distribution & Installation
- 74 Version Handling
- 78 Design & Prepare Run Schedule and Instructions
- 80 Define & Run Conversion and Implementation
- 84 Initial User Supports

- 88 Test Plan Review
- 86, 87, 89 Testing
- 92 Costs and Benefits Assessment
- 94 Handle Change Requests
- 95 Rate System/Project Continuation Alternatives
- 97 Compile Phase Report
- 99 Close the Project/Project Phase

Master Activity Plan

01 Planning

- Detailed Activity Plan

02 Assignment Dialogue

- Business & Product Quality objectives
- Requirements on system
- List of initial requirements
- Expected end products
- Time Plan
- CRD - Reference Guide

03 Follow up

13 Program Specification

- Screen dialogue
- Screen layout
 - field description
 - input fields
 - error messages
 - output fields
 - menus
- Processing Rules
 - error handling
- General error messages
- Security Rules
- Updating Rules
- Shortcut keys
- Input
 - validations
 - layouts for all input data
 - databases
 - other files
- Output
 - databases
 - other files
- List outputs
 - layout with field descriptions

25 Objectives and Constraints

40 System Functions and Structure

45 Define Business Functions and Procedures

47 Design Screen and Dialogue (create prototype?)

Detailed Activity Plan

02 Business & Process Analysis

Business & Product Quality objectives

Expected End-Products

Time Plan

CRD Reference Guide - Hur många

03 MS Project - update and report to Mark 1/week

13 Screen/Window dialogue layout

25 Create a form that can be filled in when information is provided by users

40 Functional Description - Kolla med MM

45 Write Reference Guide

47 Screens and layout
create a prototype

Problemdescription

Responsible: Ann-Christine Hagberg
Subprocess: Product Development Update

Date: 98-05-12
Version: 1.0

Problem	Cause	Consequence
The same Kit is developed twice, within, or in different, departments.	Non or insufficient sharing of information between colleagues.	Double work and extra costs.
No messaging and reporting facility.	All productdevelopers need info about changes that require an action from the user.	Errors can occur when kits that are shared is changed.
The Modelspecification is not valid.	The business is changed.	Too much information lies in footnotes.
The Applicationlist must be respecified.	The business is changed.	Too much information lies in footnotes.
The users do not like the system.	It's not user-friendly. AS/400 The Modelspecification is not valid.	The products are not manufactured. Vendors/Distributors can't find the products or modells in the system. Incomplete or not up-to date information.
Too many footnotes.	Putting in footnotes without checking if they already exists.	Too many records.
Can not list the products to the catalogues.	The system is not capable of Handling cataloguestructure.	Must update in several places.
Don't know the kits contents.	The function is missing in the system.	The updaters won't know if he entered the right product.
Seals don't exist in the system.	The function is missing in the system.	Is needed to check the kits content.

Product Development

➤ Background:

- ◆ To make one common product development application for world wide VSM, Vehicle Service Market, product division.

➤ Why product development subprocesses:

- ◆ Product development departments in fur countries with different applications and platforms.
- ◆ Difficulties to share data between departments.
- ◆ Poor security of the data.

➤ What has been done so far:

- ◆ Project onfoing since Sept. 97.
- ◆ The processes and the fields needed has been dicussed and nearly finalised.

➤ Business overview:

- ◆ What are the requiriments:
 - Product developing tool.
 - We must be able to take our work with us to an other country.
 - Consolidation of data at least once a day.
 - Facilities needed:
 - Quality, to ensure correct data entry.
 - Logging, to keep track fo history
 - Query for making reports.
 - Messaging or report.
 - Batch update.

➤ The business purpose:

- ◆ Improve quality of information.
- ◆ Immediately access to all stored information.
Increased efficiency.
- ◆ Less chance to make errors, like the same product entered twice.
Cost reduction.

➤ Specification of responsibility...

- ◆ A working team with people from Gothenburg, France and Singapore.

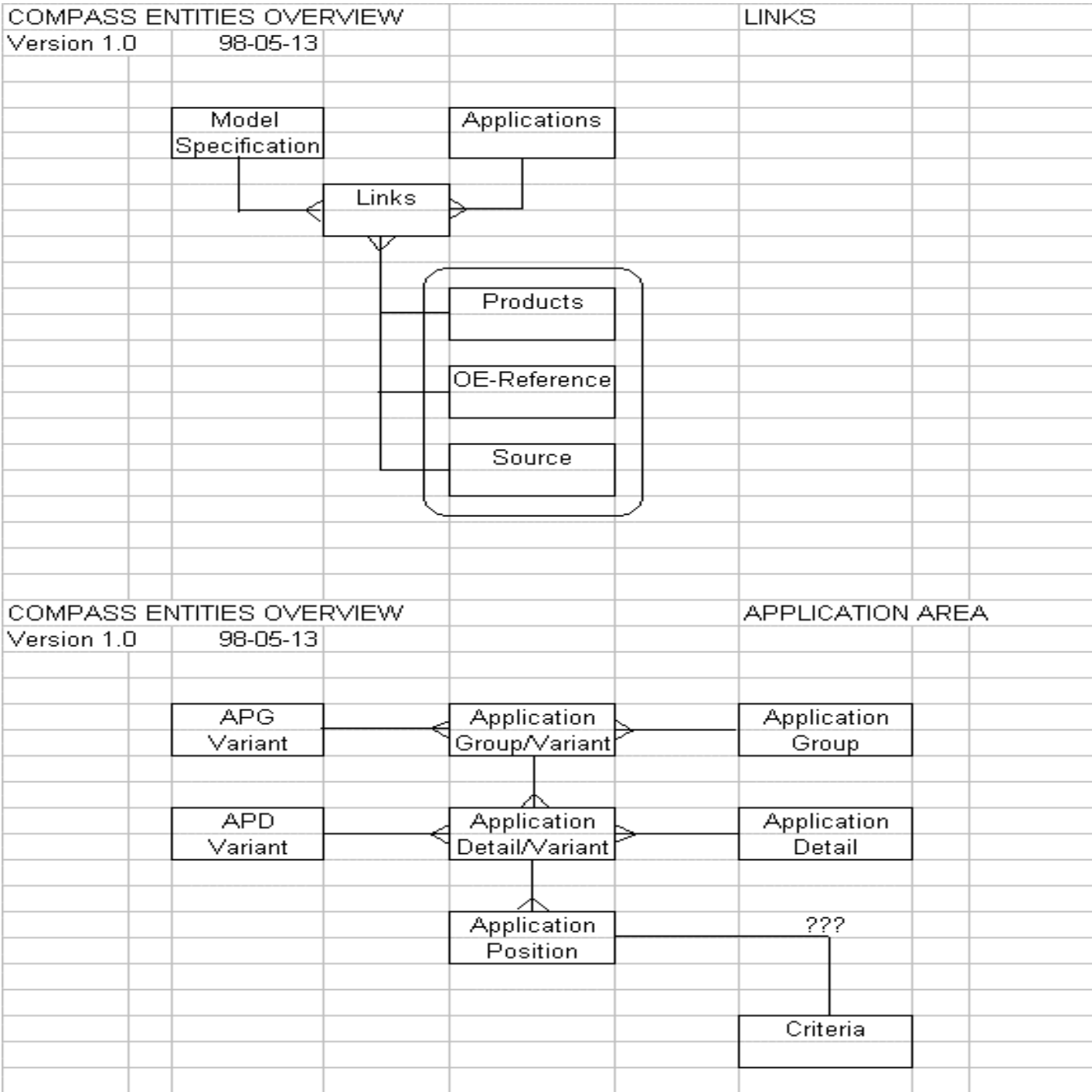
➤ Scope:

- ◆ A common application that can handle all information linked to the development of our product.
- ◆ The application linked to SKF applications like:
 - Prodmast
 - MCSS
 - MAPICS
 - Others

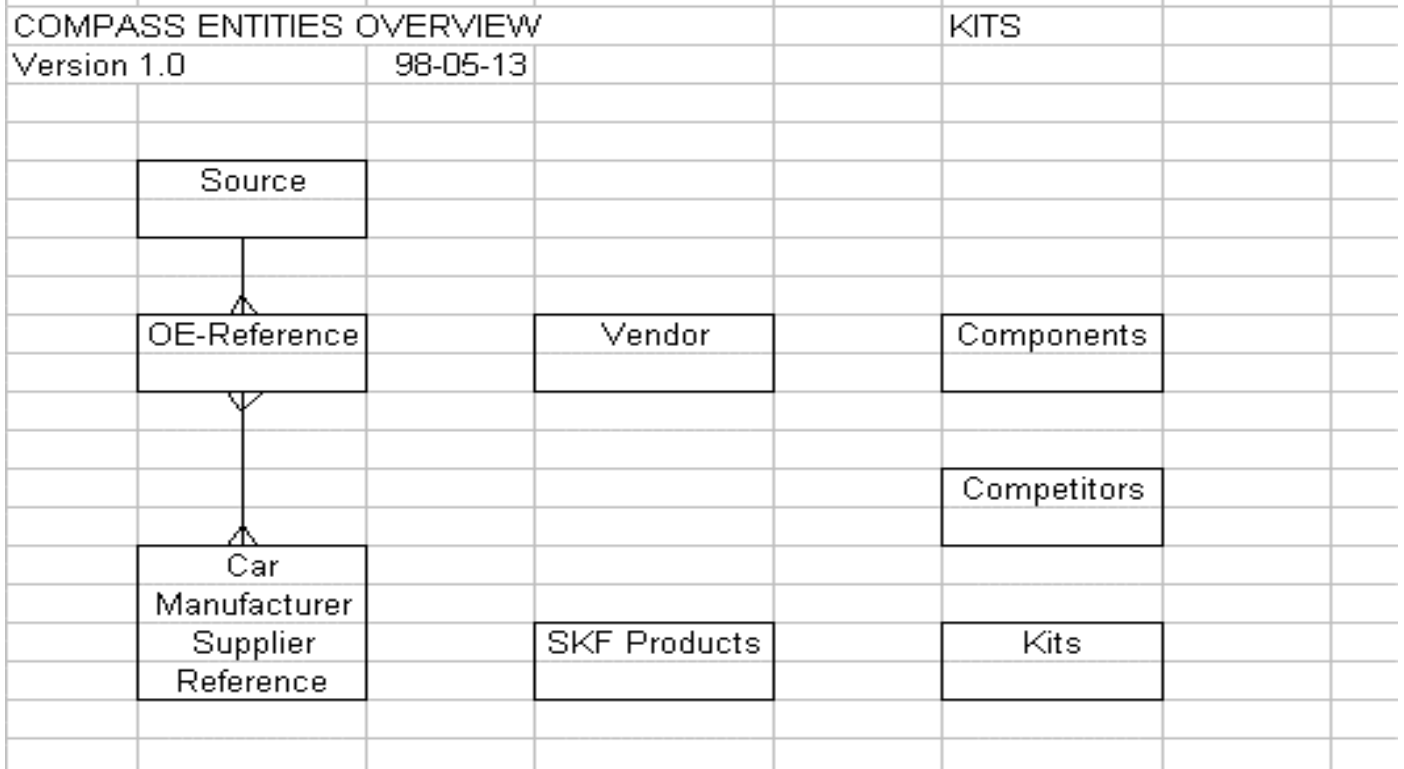
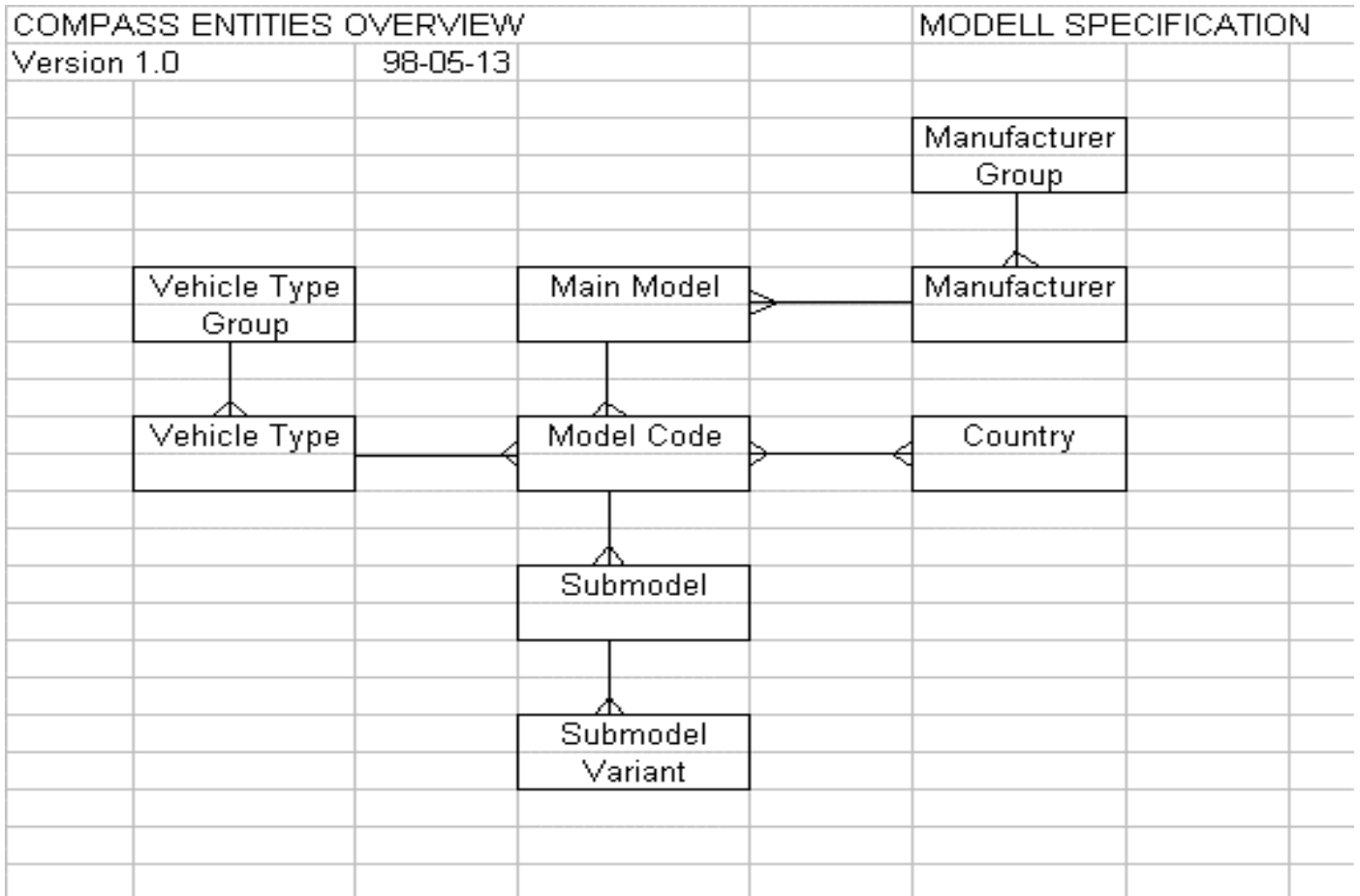
➤ Restrictions, Uncertainty & Riskfactors:

- ◆ If the Subprocess is not completed we will have difficulties to send files with the right quality to the catalogue department.
- ◆ The system must be able to provide information to other similar catalogue systems within the SKF world (I.e Australia).
- ◆ When and who will CR be involved?

Datamodellen



Bilaga 6b



Bilaga 7

Ordlista

Här följer en lista med ord och uttryck som förekommer i arbetet. Koncernspråket på SKF är engelska vilket har gjort att vissa ord och begrepp inte har gått att översätta till svenska. Även dessa presenteras här.

Batch	Buntvis inmatning
Data Dictionary	En list på alla fält och begrepp och vad de innebär och betyder.
Input	Indata, vid inmatning
Interface	Gränssnittet mellan “människa och maskin”
ISDM	SKF Information System Development Manual
Kit	Lager plus låsande eller tätande detaljer som förpackas i en box.
Online	Direktansluten
Output	Utdata, utmatningsdata
Performance	Genomförbarhet, dvs det ska gå smidigt att köra systemet.
Processing Rules	Vad som händer när användare interagerar med systemet.
Reference Guide	En systembeskrivning, beskriver <i>hur</i> systemet ska användas. Är en del av kontraktet med kunden. Där står <i>vad</i> som ska utvecklas.
Review	Officiell granskning, godkännande eller inte
Steering Comité	En liten grupp som står för de stora besluten.
VPO	Vehicle Parts Organisation, i det här fallet kunden.