

Hur går det egentligen?

om tidsuppföljning på Sandvik Systems Development

Hanna Wennerström



UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
<http://www.teknat.uu.se/student>

Abstract

Hur går det egentligen? - om tidsuppföljning på Sandvik Systems Development

How Are Things Really Going? - About Time Follow Up Reports at Sandvik Systems Development

Hanna Wennerström

This master thesis investigates the possibilities of a new Business Intelligence application at Sandvik Systems Development. The aim is to review current situation, and critical factors for such a launch, to develop a prototype and to come with further recommendations to the management team. Since Sandvik Systems Development work as a consultancy firm it is crucial to measure the time spent by the personnel, since this constitutes their entire production and thereby correlates to business revenue. Analysis of the time report system is the base for the intended business intelligence solution.

The work has been conducted through literature studies, interviews, technical design, continuous testing, and is concluded by analysis. The outcome is a comprehensive report including reviews of present situation, requirements specification, success factors and recommendations. The other outcome is a prototype designed to fulfill user and management requirements. The result is that the current system is a match of the management team's current requirements, but not necessarily their goals. Thus it will suffice for a limited amount of users and purposes. Greater possibilities are to be expected with a new application, but it comes with a development cost. A new solution could attract new users and new usage areas. The launch of a new application would require further investigation, preferably around the key factors suggested in Söderström's theoretical model and according to my recommendations.

Handledare: Linda Fernqvist
Ämnesgranskare: Roland Bol
Examinator: Elisabet Andrésdóttir
ISSN: 1650-8319, UPTec STS07 035

Populärvetenskaplig sammanfattning

På nästan varje arbetsplats finns det någon form av kontroll över hur personalen lägger sin tid. Det kan vara fastställda arbetstider, flex med stämpelkort eller förtroendearbetstid med fastställd totaltid. På Sandvik Systems Development registreras total arbetstid, men också mer detaljerat hur personalen lägger sina dagliga arbetstimmar.

Sandvik Systems Development fungerar som ett intern konsultbolag åt Sandviks affärsområden. De timmar som personalen lägger på kunder är på så sätt deras *produktion*. För att se hur det går för företaget behövs ett bra sätt att sammanställa hur personalen lägger sina timmar. Idag finns ett antal olika försök till sådan sammanställning, och examensarbetet har gått ut på att se över dessa former och se vilka förutsättningar det finns för en mer tekniskt avancerad lösning. Arbetet har också bestått i att ta fram en prototyp för att se på hur en eventuell lösning skulle kunna se ut.

Resultatet är en analys av dagsläget som åstadkommit med hjälp av intervjuer. Med detta som bas har ett kravdokument tagits fram för en framtida lösning i QlikView. QlikView är en mjukvara som används för att analysera stora mängder data. Med grund i teori har även ett antal viktiga faktorer för framgång identifierats. En prototyp i QlikView, som utarbetats, visar praktiska exempel på hur en ny lösning kan se ut, och vad den kan åstadkomma. QlikView erbjuder större möjligheter än de rapporter som finns idag, men kräver också mer arbete, och därmed kostnader. Jag tror att man i och med införandet av en QlikView-lösning kan vända sig till fler användare och få en utökad funktionalitet. Rekommendationer avslutar rapporten där jag uppmanar ledningsgruppen att, vid eventuell vidareutveckling av mjukvaran, fördela ansvaret på ett tydligt sätt, att ta hänsyn till, och förankra arbetet i verksamheten, samt kanske viktigast av allt, se till att använda slutprodukten.

Innehållsförteckning

1. Inledning	4
1.1 Problemformulering	4
1.2 Syfte	4
1.3 Uppdragsgivare och expertis	4
1.4 Disposition	4
1.5 Språkbruk	5
1.6 Lista över förkortningar och ordförklaring	6
2 Företagspresentation	7
2.1 Sandvik	7
2.2 IT på Sandvik	7
2.3 Sandvik Systems Development	8
3 Att arbeta med Business Intelligence	9
3.1 Vad är Business Intelligence?	9
3.2 Datalager	10
3.2.1 Övergripande strukturer	12
3.2.2 Normalisering	12
3.2.3 Flerdimensionella kuber	13
3.3.5 Associative Query Logic	14
3.4 Mål som utgångspunkt för Business intelligence	14
3.5 Framgångsfaktorer vid införande av Business Intelligence	15
3.5.1 Vision	15
3.5.2 Struktur	16
3.5.3 Process	16
3.5.4 Teknik	16
3.5.5 Individ	16
3.5.6 Organisation	17
3.6 Sammanfattning	17
4 Metod	18
4.1 Om arbetet från problem till lösning	18
4.2 Perspektiv på teorin	18
4.3 Att utreda behov och användare – intervjuförfarande	19
4.4 Avgränsningar	19
4.5 Sekretess	20
5 Beskrivning av nuläge	21
5.1 Bakgrund	21
5.2 Hur produceras rapporterna idag?	21
5.3 Vem använder rapporterna?	22

5.4 Struktur och nomenklatur	22
5.5 Hur ser dagens rapporter ut?	24
6 Översikt av programvara	28
6.1 IFS	28
6.2 Projus	28
6.3 SAP Business Information Warehouse (SAP BW)	28
6.3.1 Laddning av data till datalagret	28
6.3.2 Datalagret	28
6.3.3 Rapportering	29
6.4 SAP-connector	29
6.5 QlikView	29
7 Förutsättningar för en ny lösning	30
7.1 Stjärnstrukturer i praktiken	30
7.2 Underrättelsecykeln i praktiken	31
7.3 Förundersökning	31
7.3.1 Problem med dagens rapporter	31
7.3.2 Lösningar på dessa problem	31
7.4 Kravspecifikation	32
7.4.1 Funktionell beskrivning	32
7.4.2 Konsekvenser	32
7.4.3 Tolkning av användarkrav	32
7.5 Kritiska framgångsfaktorer i praktiken	32
7.5.1 Vision	32
7.5.2 Struktur	33
7.5.3 Process	33
7.5.4 Teknik	33
7.5.5 Individ	34
7.5.6 Organisation	34
8 Prototypen	35
8.1 Design och innehåll	35
8.1.1 Datakopplingar	35
8.1.2 Layout	35
8.1.3 Att göra sökningar	37
8.2 Funktioner	38
8.2.1 Tidskategorier	38
8.2.2 Drill-down på önskad storhet	38
8.2.3 Ett antal olika visningsmöjligheter	39
8.2.4 Budgetjämförelse	40
8.2.5 Extern debiteringskvot	40
9 Diskussion	41
9.1 Prototypens styrkor	41
9.2 Prototypens brister	41

9.3 Tidsrapporteringssystemet som allt bygger på	41
9.4 En komplicerad verklighet	42
9.5 Fördelar med en framtida QlikView-lösning	42
9.6 Nackdelar med en framtida QlikView-lösning	43
9.7 Ekonomiska aspekter	43
10 Sammanfattande rekommendationer	44
10.1 Rekommendation till Sandvik Systems Developments ledningsgrupp	44
10.2 Följder vid ett eventuellt införande	44
11 Källförteckning	45
11.1 Publicerat material	45
11. 2 Opublicerat material	45
11.3 Elektroniska källor	45
11.4 Muntliga källor	45
11.5 Figurförteckning	46
11.3.1 Källförteckning för figurer	46
12 Bilagor	47
Förteckning av bilagor	47

1. Inledning

På Sandvik Systems Development sammanställer man kontinuerligt rapporter om hur verksamheten löper, och jämför dessa mot mål och budget. En av faktorerna man är intresserad av är hur personalen spenderar sin arbetstid. Detta är mycket viktigt att hålla reda på eftersom företaget fungerar som ett konsultföretag, vilket innebär att personalens timmar motsvarar företagets produktion. Hur man kan hålla koll på företagets produktion, alltså personalens arbetstimmar kommer att granskas i detalj i detta examensarbete.

I nuläget sker detta på ett antal olika sätt, men alla har sin bas i de tidsrapporteringssystem som finns till hands. Föreliggande utredning kommer att ta upp på ett antal olika faktorer kring hur den rapporterade tiden presenteras, vilka rapporter som finns, vad dessa består av, var informationen hämtas, vad den används till och hur man kan förbättra den.

1.1 Problemformulering

Sandvik Systems Development använder idag ett antal olika tidsrapporter som alla har olika styrkor och brister. I mitt examensarbete utreder jag möjligheten att skapa en gemensam lösning, som kombinerar styrkorna från nuvarande rapporter, och samtidigt försöker lösa några av de problemen som finns med dagens rapporter. Jag undersöker vilka krav som bör ställas på en sådan gemensam rapport samt hur en mjukvarulösning kan svara mot dessa krav. Jag har även utvecklat en prototyp i QlikView¹ för att ge en bild av hur denna lösning kan se ut.

1.2 Syfte

Syftet med examensarbetet är

- i) att utreda bakgrunden och förutsättningarna för en eventuell QlikView-applikation
- ii) att utveckla en prototyp i QlikView
- iii) att komma med rekommendationer för vidare arbete

1.3 Uppdragsgivare och expertis

Uppdraget har tillkommit på beställning från delar av Sandvik Systems management team och utförts inom den avdelning som jobbar med Business Intelligence på Sandvik Systems Development. Beställarna har inte varit involverade i det dagliga arbetet under uppdraget. Det dagliga arbetet har hanterats av personal från avdelningen, främst Linda Fernqvist, som arbetar med SAP BW. Jag har också fått ovärderlig hjälp av Lena Josefsson, som arbetar som QlikView-expert på Climber och Erik Pihl på DAF har fått mig att förstå den rapportstruktur som finns på företaget idag. Flera andra har också bidragit med experthjälp inom sitt område.

1.4 Disposition

Under *Företagspresentation* ger jag en översyn av Sandvik och hur IT-verksamheten är koordinerad. Jag presenterar även min uppdragsgivare, Sandvik Systems Development.

¹ en mjukvara för analys, förklaras senare

Att arbeta med Business Intelligence är den teoretiska grunden för hur man kan tänka runt Business Intelligence. Avsnittet ger en referensram till viktiga begrepp, och presenterar analysverktyg som jag senare valt att ta hjälp av för att presentera mina resultat.

Beskrivning av nuläge är en analys av läget kring de rapporter som finns idag.

I *Metod*-delen resonerar jag kring hur jag valt att gå till väga under arbetets gång samt vilka praktiska problem som jag tagit ställning till

I *Förutsättningar för en framtida lösning* har jag försökt sammanställa det undersökande arbete jag utfört på ett så konkret sätt som möjligt. Eftersom arbetet varit abstrakt på många sätt har jag valt att ta hjälp av de presenterade teoretiska analysmodellerna.

Avsnittet *Prototyp* beskriver den prototyp jag tagit fram.

Diskussion är mitt sätt att problematisera kring mina resultat. Här diskuterar jag följder och problem kring ett eventuellt skarpt införande av systemet.

Sammanfattande rekommendationer finns sist i arbetet och består av konkreta råd och slutsatser till utrednings beställare, alltså Sandvik Systems Developments ledningsgrupp.

1.5 Språkbruk

Mycket av det material jag tar upp är datorrelaterade termer. Och de flesta sådana är på engelska. Jag har skrivit rapporten på svenska, men det har på många ställen smugit sig in ”datorsvenska” i texten. Exempel på detta är trigga, query, default och in house. Målgruppen för min text är dels någon med kunskap motsvarande STS-programmets studenter och dels personalen på SSD. Det är min uppfattning att min målgrupp förstår dessa typer av engelska datortermer. Jag har samlat ord och förkortningar som jag uppfattar som antingen utanför målgruppens kunskapsområde eller som särskilt viktiga för förståelse av uppsatsens innehåll i en ordlista i stycket nedan.

Vissa ord används synonymt i texten för att få ett varierat språk. Begreppet datastrukturer används både allmänt och syftar på den struktur som data har (hur det är format, ihopkopplat etcetera) och specifikt i teori och analys som syftandes på en beskrivning av en särskilt struktur (exempelvis stjärnmodell som är synonymt med kubstruktur). Det kategorisystem som utvecklats tidstyper benämns som tidsuppdeleningen eller tidskategorisystemet och finns bifogat i en bilaga (som är borttagen i denna version). Ordbudet kring de rapporter som finns kan också verka förvirrande. Det finns huvudsakligen fyra olika rapportformer som nämns; Bex-rapporter, två sorters Excel-rapporter samt min QlikView-prototyp. Jag har försökt att vid olika tillfällen som de nämns tydliggöra vilken rapport som avses, men här kommer ytterligare förtydningar. Då jag skriver Excel-rapporten avses den rapport som produceras av DAF om jag inte specifikt skriver något annat. QlikView-rapporten benämns även som prototypen och applikationen.

1.6 Lista över förkortningar och ordförklaring

1NF	första normalformen
2NF	andra normalformen
3NF	tredje normalformen
AQL	Associative Query Language
BCNF	Boyce-Codds normalform
BEx	Business Explorer, ett rapportverktyg som finns i SAP BW. Detta är även ett av de befintliga rapportverktygen på Sandvik Sysytems Development idag. Det finns två aktuella BEx-rapporter aktuella för mitt arbete, benämnda <i>Scorecard</i> och <i>Project Follow Up</i> .
BI	Business Intelligence, begreppet förklaras i kapitel 3
BW	Business Warehouse är SAPs data warehouse-miljö
DAF	Team code för den gemensamma ekonomiavdelningen på Sandvik Systems Development och Sandvik Information Technology
drilla	det engelska ordet för att borra används för att beskriva sökning ned i datamängden för att hitta de specifika detaljer som efterfrågas
ER	Entity Relationship
ERM	Entity Relationship Modeling
idle	det finns ingen helt träffande svensk översättning; <i>overksam</i> , <i>utan syfte</i> , <i>tekniskt stillastående</i> och <i>att gå på tomgång</i> är några exempel. I texten används ordet oöversatt som en kategori av tid då man inte är allokerad till en särskild uppgift, alltså icke-produktiv tid.
OLAP	Online Analytical Processing är den beskrivande term som används för komplex analys av det data man hämtar från ett datalager
QV script	QlikView, programmet beskrivs i texten, i genomgången av programvara ett script är en textfil som innehåller kommandon till ett program eller operativsystem. I texten används ordet om den textfil i QlikView där man skriver kod som sedan exekveras då scriptet laddas
SSD	Sandvik Systems Development
SIT	Sandvik Information Technology
SAP	Systems Applications and Products in Data Processes, SAP, är idag ett av världens största affärssystem med ett stort antal olika funktionsmoduler
team code	På SSD tillhör medarbetarna ett team. Detta team benämns med en bokstavskombination, där alla på SSD-team börjar med bokstaven L, men annars utan större logik . Antalet bokstäver markerar hierarkin, exempel LC (Competence) ligger över LCF (Microsoft). Varje team har en kompetenschef (även benämnd teamchef), vars engelska titel är team manager
top down- bottom up	strategi som innebär att man antingen börjar från detalj- eller helhetsperspektiv och jobbar sig sedan antingen <i>upp</i> från detaljerna eller <i>ned</i> från helheten

2 Företagspresentation

Jag har skrivit examensarbetet för Sandvik Systems Development (SSD). Eftersom deras kunder enbart utgörs av andra Sandvikbolag och eftersom att mycket på bolaget påverkas av koncernen i stort har jag valt att, förutom en beskrivning av SSD, ge en kort introduktion även till Sandvik i stort.

2.1 Sandvik

Sandvik har funnits sedan 1862 och är en verkstadskoncern med produkter som är världsledande position i sina branscher. Sandvik finns representerat i 130 länder över hela världen och har ca 42 000 anställda. Hela Sandvik Group omsatte 2006 ca 72 miljarder kronor.

Sandviks kärnverksamhet är olika typer av materialteknik inom tre huvudområden, som var och en representeras av ett eget affärsområde:

- Verktyg i hårdmetall och snabbstål för metallbearbetning samt ämnen och komponenter i hårdmetall. Bolaget som framställer dessa är Sandvik Tooling.
- Maskiner och verktyg för bergavverkning och gruvdrift, vilket är Sandvik Mining and Constructions verksamhetsområde
- Rostfria och höglegerade stål, specialmetaller, motståndsmaterial samt processsystem. Dessa material utvecklas och produceras av Sandvik Material Technology.

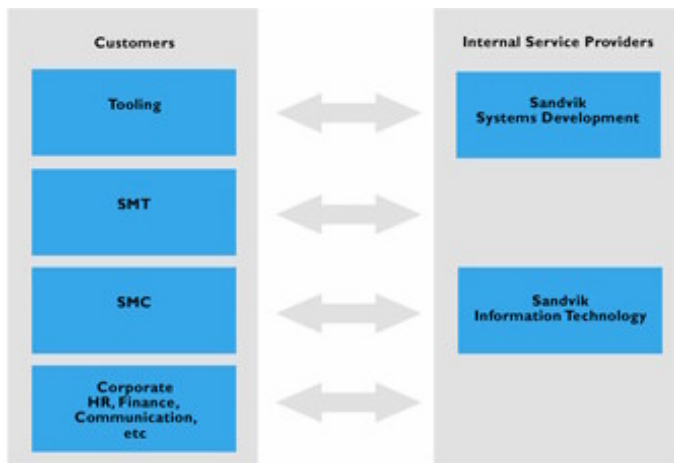
(Sandviks hemsida, 2007-08-13)

2.2 IT på Sandvik

Sandvik har två fristående bolag för att utveckla, leverera och sköta driften av företagets IT-lösningar. Sandvik Information Technology har hand om hårdvara, driftsfrågor och service. Sandvik Systems Development utvecklar mjukvara och producerar interna systemlösningar.

Utöver detta har de tre producerande bolagen egen IT-verksamhet, som oftast fungerar som beställarsida gentemot IT-bolagen.

All IT-verksamhet inom Sandvik Group koordineras av företagets Chief Information Officer. De tre affärsområdena (Tooling, SMT och SMC) har varsin respektive CIO som är underställd honom, och det är även VD:arna för IT-bolagen Sandvik Information Technology och Sandvik Systems Development. (The Sandvik IT Portal, 2007-08-13)



Figur 1- IT-koordineringen på Sandvik

2.3 Sandvik Systems Development

Sandvik Systems Development (SSD) är Sandviks systemutvecklingsbolag. SSD har runt 350 anställda varav 210 i Sandviken och 5 i Gimo, resten är utplacerade på SSD-kontor världen om.

SSD:s uppgift är att i nära samarbete med koncernens affärsområden och bolag stödja dem med verksamhetsnära och innovativa IT-utvecklingar. Bolagets verksamhet omfattar analyser och utredningar, projektledning, systemutveckling, implementation och förvaltning av system. De arbetar i en internationell miljö med Sandvikprojekt över hela världen. SSD fungerar som ett konsultbolag, och har Sandvikbolagen som enda kund.

SSD har nyligen ändrat hela sin organisationsstruktur till en så kallad *matrisorganisation*. Det innebär att organisationen är strukturerad i två dimensioner, dels är personalen uppdelad i team, efter kompetens, som leds av en teamchef. Utöver detta är man allokerad till ett aktuellt projekt eller leverans där det finns en leveranschef. Vissa verksamheter faller utanför den ramen, och dit hör exempelvis övergripande funktioner som HR, Sales & Marketing, vissa servicefunktioner samt ledningsgruppen. (SSD:s intranät, 2007-08-13)

3 Att arbeta med Business Intelligence

Under denna rubrik ska jag redogöra för ett antal olika faktorer kring Business Intelligence och hur man kan arbeta med det. Jag börjar med att redogöra ett antal grundläggande frågor om *vad* data, information och Business Intelligence är, och vad som är relationen mellan begreppen. Informationen som ligger till grunden för Business Intelligence måste tänkas igenom innan man börjar arbeta med det praktiska arbetet. En modell för hur man kan lägga upp det analysarbetet presenteras genom en så kallad underrättelsecykel.

I de flesta fall inhämtas informationen som används från någon typ av lagringskälla. Eftersom det ofta rör sig om stora datakällor med olika faktorer och dimensioner (som kommer sig av att de försöker beskriva komplexa system, som exempelvis en hel organisation eller en hel leveransprocess) så använder man sig ofta av ett lagringsmedium kallat ett datalager (eng. *data warehouse*). Ett datalager har nämligen lagringsstrukturer för mer komplex data än en vanlig databas. Jag beskriver därför olika begrepp som gäller lagring och modellering av datalager.

Det är även viktigt att förstå *varför* man använder sig av Business Intelligence. Den vanligaste och viktigaste anledningen är att det ger ett konkret och strukturerat sätt att arbeta runt de mål som finns med verksamheten. Därför behandlar jag målarbete i nästa avsnitt. Då ges konkreta tips på hur mål ska vara utformade så att de blir effektiva. Detta är ett mycket viktigt grundarbete som är helt avgörande för verksamheten som bedrivs och något som bör behandlas innan man börjar med implementeringen av Business Intelligence då BI-systemet kommer att vara avhängigt av att målen är tydligt formulerade.

När man valt att införa ett tekniskt Business Intelligence-system finns det en hel del faktorer som är viktiga att tänka igenom för att undvika fallgropar. Peter Söderström har tagit fasta på ett antal sådana faktorer. Hans modell gäller för införandet av ett datalager, men jag uppfattar en stor del av problematiken som jämförbar med införandet av ett BI-system, både därför att de tekniska förutsättningarna är snarlika och för att processen sker på liknande ställen i organisationen.

3.1 Vad är Business Intelligence?

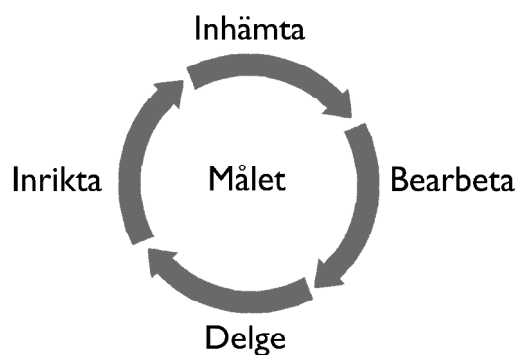
För att förstå Business Intelligence i ett sammanhang beskriver jag relationen mellan data, information, och intelligence. Intelligence kan ungefär översättas med underrättelse och då man använder det för att samla och presentera information i affärssyften så kallas det Business Intelligence. Detta har blivit så vanligt att Business Intelligence blivit ett vedertaget begrepp.

Data innebär obehandlade och neutrala utsagor om verkligheten som kan samlas in eller registreras på olika sätt. Dessa utsagor blir i sig själva inte särskilt intressanta. Men om dessa data arrangeras eller sätts i ett sammanhang kan de bli meningsfulla för en person vid ett specifikt tillfälle. Då övergår datat till information. Man kan alltså beskriva det som att information ligger till grund för ett bestämt mål. Eftersom detta beror på sammanhang kan det vara så att det som är information vid en tidpunkt för någon kan vara meningslösa data för någon annan. (Saxby 1990, s 11)

Holopainen, Lillrank och Paavola menar att information består av tre olika faktorer. *Innehållet* är den substansen av fakta som informationen innehåller, *formen* är de signaler, symboler och format som för fram fakta och *kontexten* är den specifika situation som fakta framförs i som information.

För att uppnå ytterligare en nivå av meningsfullhet används begreppet intelligence (vissa delar av litteraturen använder begreppet kunskap som nästa nivå efter information). När olika delar av information filtreras och analyseras för att uppnå förståelse uppnås intelligence. Detta kan exempelvis ligga till grund för beslut eller analyser av olika slag. (Kahaner 1996, s 21) Ett svenskt ord som används är underrättelse och många av de grundläggande idéerna om intelligence är hämtade från den militära världen och många exempel i litteraturen kring ämnet är från krigshistorien. När tillvägagångssättet används inom ett företag för att på olika sätt skaffa sig kunskap brukar det benämnas *Business Intelligence*. Detta kan användas inom en rad olika områden som exempelvis att kartlägga konkurrenter, att kontrollera den egna produktionskedjan, säljframgångar, eller som i fallet för denna utredning; för att följa upp intern produktivitet.

Sundström beskriver arbetet som en underrättelsecykel där allting kretsar kring uppsatta mål. Underrättelsecykeln är ett begrepp som använts länge inom militär underrättelseverksamhet för att konkretisera hur flödet och analysen kring information bör se ut, och modellen finns i ett antal olika varianter. I den modell Sundström presenterar har cykeln fyra delfunktioner som är att inrikta, inhämta, bearbeta samt att delge information. (Sandström, 2004, s 135ff)



Figur 2 – Underrättelsecykeln

Under arbetet med informationshantering i underrättelsecykeln ligger målet tydligt i mitten, och utgör en utgångspunkt. Detta är ett tecken på hur viktigt det är att arbeta med målformulering. Målformulering tas upp i ett senare stycke.

3.2 Datalager

För att kunna genomföra de olika stegen i underrättelsecykeln måste man ha ett medel att lagra inhämtad data. Det vanligaste sättet att lagra data i allmänhet är i en databas, som är

en samling av lagrad, relaterad data. Men då det gäller Business Intelligence är det vanligare att lagra i ett så kallat datalager².

Den generella skillnaden mellan en databas och ett datalager (eng *data warehouse*) är att ett datalager har en mer komplex struktur med flera dimensioner. Detta innebär flera praktiska skillnader, som kan ses i Inmons definition nedan. Det innebär också att ett datalager ofta är mer passande för OLAP (Online analytical processing), beslutsstödsfunktioner och olika typer av trendanalyser. Eftersom denna typ av funktioner är mer komplexa än vad som krävs av en databas krävs också att datalager har en effektivare tillgänglighet till datamängderna (eng *query support*). (Elmasri & Navathe 2004, s 900f och s 910)

Bill Inmon, en av förgrundsfigurerna inom utvecklingen av datalager, definierade redan 1992 begreppet så här: *a data warehouse is a subject oriented, integrated, non-volatile, and time variant collection of data in support of managers decision.*

Där de specifika orden definieras (egen översättning, med inspiration från Forsberg & Morell 2006):

- Subjektorienterad: Ett datalager fokuserar på subjekt istället för dagliga transaktioner
- Integrerad: Ett datalager konstrueras genom att varierande källor integreras till en sammanhållen enhet
- Icke ombytlig: Ett datalager ska innehålla stabil data. Data tillförs men tas inte bort eller ändras. Detta bidrar till en konsekvent förståelse
- Tidsberoende: All data i datalagret är förknippad med en tidsperiod

Definitionen står sig fortfarande väl trots att den formulerades för ett och ett halvt decennium sedan, förutom de detaljer som ändrats genom att lagringskraven ökat så enormt de senaste åren. Detta innebär att man i vissa fall kan bli tvungen att ta bort data, så beständighetskravet kan ibland inte längre uppfyllas i sin helhet.

Inmons definition är mycket tydlig och exakt. För att ge lite perspektiv till definitionen vill jag också nämna en mer praktiskt orienterad definition. Jag har i mitt arbete använt mig av en analysmodell från Söderström (1997), och han arbetar med en egen definition, som ligger till grund för hans modell: *ett datalager är en logiskt sammanhållen datamängd, som är avsedd för analys och som speglar flera tidsperioder genom att data regelbundet hämtas från andra register.* Denna definition är mer inriktad på användandet och hur man praktiskt har nytta av ett datalager. Jag har inte själv tagit fram något datalager i detta arbete men som reflektion kan sägas att definitionen också speglar funktionen i den prototyp som tagits fram eftersom prototypen fungerar som logiskt sammanhållen datamängd som är avsedd för analys. Den färdiga applikationen är tänkt att spegla data från flera tidsperioder genom att data regelbundet ska hämtas från andra datalager. Detta är ett argument för att Söderströms teorier är befogade att applicera inte bara vid arbetet med ett datalager, utan även då man pratar om en teknisk Business Intelligence-lösning.

² Översättningen från engelska är något otydlig eftersom det svenska ordet lager har två möjliga betydelser. Jag vill poängtera att datalager syftar på att det är ett "förråd" med data och inte ett "skikt", som man kan förledas att tro.

3.2.1 Övergripande strukturer

När man skapar ett datalager måste man ta hänsyn till ett antal faktorer gällande modelleringen av datamängderna. Det gäller helt enkelt hur man ska kunna hantera all data på ett praktiskt sätt. Denna modellering kan vara på olika plan, logiskt, konceptuellt eller fysiskt, och alla dessa tre sätt är nära besläktade.

Ett datalager och en databas är i grunden inte särskilt olika, förutom med avseende på komplexitet och vissa modellval. Jag beskriver inledningsvis två strukturer för databaser. ERM-modellen är den mest utbredda och länge har utgjort grunden för hur man konceptualiserar databaser. Den mjukvara som jag använt, QlikView, gör då den laddas, om tidigare strukturer till en associativ datastruktur för att optimera prestanda.

Ett av de mest kända sätten att strukturera data är med Entity-Relationship Modeling (ERM). ERM är en logisk teknik som är tänkt att ta bort eventuell redundans, den bygger på att alla entiteter i modellen skapas med relationer sinsemellan. Det är ett sätt att strukturera data som är detaljrikt och logiskt. Nackdelarna är dock att om komplexiteten på datat ökar så blir ER-diagrammen väldigt komplexa och svåröverblickbara. Eftersom datalager per definition är stora och komplexa så blir ER snabbt en svårhanterlig modell. Det kan bli så komplext att det inte finns möjlighet för användaren att överblicka de olika relationerna. (McCarthy & Risch 2005, s 27ff)

En associativ databas har försökt optimera problemen med relationsdatabaser för att vinna tid och effektivitet. Posterna i databasen lagras som objekt och associationer. Objekt är oberoende av varandra och associationerna beskriver relationerna dem emellan.

Ett illustrerande exempel kan vara om man vill föra register över de anställda på Sandvik Systems Development och deras kompetenser. I en relationsdatabas krävs tre tabeller; en för de anställda, en för kompetenser och en för vem som har vilken kompetens. I en associativ databas räcker det med två; en för anställda, en för kompetens samt en dubbelriktad association för *har kompetens inom*. (Strömberg 2006, s 11ff)

3.2.2 Normalisering

För att få en databasmodell som är så smart och icke-redundant som möjligt pratar man om begreppet normalisering. Det är ett sätt att dels få tydliga tabellstrukturer och dels att effektivisera lagringen i databasen. Grundregeln för normalisering är att det ska finnas "en typ av sak per tabell och en sak per rad" (McCarthy & Risch 2005, s 217). För att tydliggöra hur normaliserad en datastruktur är finns det standarder för normalisering. De vanligaste är 1NF, 2NF, 3NF och BCNF där NF betyder normalform och BCNF betyder Boyce-Codds normalform. BCNF är den striktaste formen av de fyra, det vill säga att den ska uppfylla flest kriterier.

Kortfattat kan man säga att de nämnda normalformerna bygger på kriterier för att få en så entydig struktur som möjligt. 1NF innebär att det endast får finnas atomära värden, alltså ett värde per ruta och tabellrad. 2NF måste uppfylla 1 NF samt att de attribut som ej är nycklar ska vara fullständigt funktionellt beroende av tabellens möjliga nycklar (kandidatnycklar).

Med fullständigt funktionellt beroende menas att hela nyckelkombinationen [A] entydigt bestämmer ett annat attribut [B]. Detta kallas att [A] är en minimal determinant till [B].

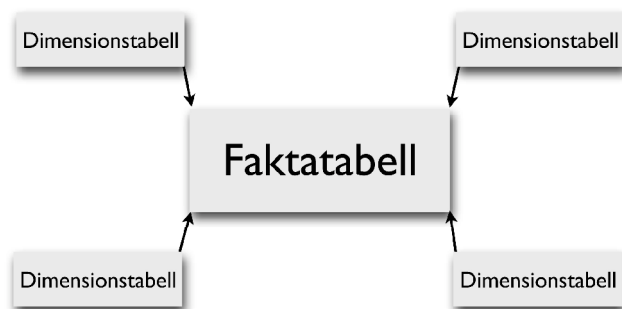
För 3NF ska 2NF vara uppfyllt samt att de attribut som inte är nycklar inte får vara fullständigt funktionellt beroende av något annat attribut som inte heller är en nyckel. BCNF ska uppfylla 3NF samt att nyckelattribut inte får vara fullständigt funktionellt beroende av icke-nyckelattribut.

I databaser normaliseras tabeller vanligen till 3NF eller BCNF för att motverka redundans och för att man bara behöver ändra på *ett* ställe vid uppdatering. Skillnaden mellan sökningar i databaser och datalager är att i en databas behövs oftast bara en post (ex. ”vad heter kund nummer X?”) medan en sökning i ett datalager kan behöva många poster (ex. ”vilken användare har rapporterat tid på projekt Y under månad Z?”). Därför optimerar man ett datalagers struktur med tanke på utsökningar snarare än uppdateringar. Detta innebär vanligen att strukturen är *denormaliserad*, som alltså innebär att man inte uppfyller någon eller några av normaliseringskriterierna. Det finns även flerdimensionella strukturer som kan vara normaliserade (exempelvis snöflingestruktur, se figur 4). (McCarthy & Risch 2005, s 217ff)

3.2.3 Flerdimensionella kuber

Datamängderna i lagret måste struktureras på något sätt för att det ska bli överblickbart samt gå att hantera. Datat struktureras i olika grupper, nivåer eller tabeller med nycklar emellan för att åskådliggöra hur det verkligen ser ut. Modellval kan bero på hur datat ser ut. Det finns olika strukturer, två vanliga är stjärnstrukturer och snöflingestrukturer.

Stjärnstrukturer innebär att det finns en faktatabell där faktiska transaktioner registreras, och till denna ”avskalade” faktatabell är det kopplat ett antal dimensioner i form av tabeller med data³. (Elmasri & Navathe 2004, s 905)

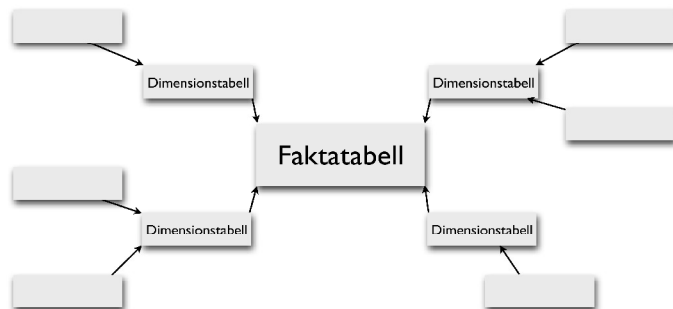


Figur 3 - Schematisk skiss av en stjärnstruktur

³ För att exemplifiera så utgör de faktiska timrapporteringarna, ”X timmar under dag Y”, transaktionerna i tidrapporternas faktatabeller. Dimensionerna utgörs av all ”kringdata”, som personalinformation, jobbtitlar, vilka debiteringstaxor som finns etc. För detaljer, se kapitel 7.1 samt bilaga 5.

Det finns även andra varianter så stjärnstrukturer, till exempel en så kallad värdekedja som är en sammankoppling av ett antal stjärnstrukturer som är sammanfogade via gemensamma dimensionstabeller. (Söderström 1997 s 66)

Snöflingestrukturer bygger på samma princip som stjärnstrukturer, med en faktatabell för transaktioner i mitten, men skillnaden är att de dimensioner som kopplar in mot faktatabellen är normaliserade, och det kan därför finnas flera, organiserade hierarkiskt. (Elmasri & Navathe 2004, s 905)



Figur 4 - Schematisk skiss av snöflingestruktur

3.3.5 Associative Query Logic

I QlikView används en annan datastruktur än dimensionsmodellerad information. Associative Query Logic (AQL) är QlikViews teknik för att arbeta med stora datamängder och behandla sökningar (queries) blixtnabbt. Under laddningen till programmet omstruktureras datamängderna från kubstrukturen till ett så kallat datamoln. Datamolnet bygger på icke-redundanta tabeller som hålls i minnet. Queries görs sedan direkt mot datamolnets minnesbank. Strukturen är minneseffektiv eftersom alla data sparas bara en gång och associationerna lagras som pekare. Det kräver ingen indexering, register eller fördefinierad analys eller queries, allt kan köras direkt i applikationen. (Strömberg 2006, s 11ff)

3.4 Mål som utgångspunkt för Business intelligence

Enligt March (1999, s 75) kan man definiera organisationers lärande som rutinbaserat, historieberoende och målorienterat. En BI-lösning är ett sätt för en organisation att lära och skapa en chans till proaktivt agerande. Detta gör man genom att försöka hitta en applikation som är just rutinbaserad och historieberoende samt, viktigast av allt, målorienterad.

Vikten av målorientering är något som nämns i nästan varenda källa kring lärande, projekt och IT-system, och jag har därför valt att ta upp det i ett separat stycke. De tidskategorier, budgetmål och debiteringsgrader som bestämts av framför allt SSD:s ledning är en typ av verksamhetsmål. För att ge en bredare bild än enbart den tekniska vill jag även föra in dimensionen kring rapporternas faktiska syfte. Eftersom dessa syftar till att ge ett verktyg för måluppföljning så är det viktigt att ge en bakgrund till hur man ser på mål och måluppföljning, och hur Business Intelligence kommer in i det resonemanget.

Jag har på många sätt försökt visa vikten av att sätta upp mål för arbetet och att det kommer att påverka en framtida BI-lösning. Mål står som en faktor i mitten av underrättelsecykeln och organisationens lärande (som styr om lösningen är till nytta eller ej) beror av om applikationen är målorienterad. Med detta sagt så är det tydligt att målen måste formuleras *innan* arbetet med själva BI-applikationen kan ta sin början. Det är alltså viktigt att ha en tydlig målbild som utgångspunkt för att kunna arbeta på ett bra sätt med Business Intelligence. Att styra verksamheten baserat på mål har blivit en grundregel för ledarskap. Tanken har i modern verksamhetsstyrning fått benämningen *MBO- Management by Objectives*, och är ett sätt att kontinuerligt följa upp och utvärdera den verksamhet man är utsedd att leda (Anthony & Govindarjan 2007, s 136, 386f, 520).

För att de mål man ska ta fram (och sedan ha som utgångspunkt i framtagandet av en BI-lösning) ska fungera och vara effektiva finns det en konkret modell över hur målen ska vara utformade. Sandström (2004) menar att om målen man sätter upp för verksamheten ska vara lyckade så bör de vara SMART, alltså *specifika, mätbara, accepterade, realistiska och tidsbestämda*. Lite mer utförligt så innebär detta att målen bör vara entydiga och situationsanpassade (specifika), man ska kunna läsa av utfall och riktning (mätbara), det som är tänkt att uppfylla målet måste godta detta (accepterat), det ska vara möjligt att nå målet (realistiskt), samt att det är viktigt att veta när allting ska vara uppfyllt (tidsbestämda). (Sandström 2004, s 54ff)

3.5 Framgångsfaktorer vid införande av Business Intelligence

Det finns mycket som kan gå fel vid införandet av ett tekniskt system. Risken att misslyckas blir större ju större del av verksamheten som berörs. Eftersom en BI-lösning har potential att beröra många viktiga punkter i organisationen är det viktigt att få en bra överblick över vad som kommer att avgöra framgången hos en framtida applikation. Peter Söderström tar i sin bok "Datalager - verksamhet, metod, teknik" upp en modell bestående av sex olika dimensioner som man bör beakta vid implementering av ett datalager. Jag har valt att beskriva modellen eftersom jag tycker att det utgör ett handfast verktyg att beskriva dimensioner runt en teknisk lösning. Med dessa dimensioner inkluderas många av de faktorer som är relevanta för att göra en bedömning av den BI-applikation som hela studien kretsar kring. Detta ger ett applicerbart verktyg att jobba med. Viktigt att poängtera är också att de system jag granskat inte rört sig om en extern informationshantering utan intern, vilket gjort att jag valt att utesluta de delar som rör just externa faktorer, exempelvis marknadsfaktorer, kundpreferenser och liknande.

3.5.1 Vision

Ett datalager är strukturerat för att stödja beslutsfattande på något sätt. Därför är det viktigt att tillhandahålla de data som potentiellt är viktiga för styr- och ledningsfunktioner. För att göra detta måste man veta vad ledningens visioner är och de visioner som finns med verksamheten.

Kritiska framgångsfaktorer för vision:

- Knyt datalager till prioriterade mål och strategier
- Visa omedelbart synliga vinster

- Inse hur verksamheten kan utnyttja nyttan med datalagret

3.5.2 Struktur

För att kunna utnyttja ett datalager till fullo är det viktigt att ha en bra struktur. Strukturen är det viktigaste verktyg för att kunna extrahera intressant och relevant information. Strukturen bör spegla det sätt som beslutsfattarna ser på verksamheten. För att lyckas med detta är det viktigt att kravställare och konstruktörer tillsammans jobbar fram en fungerande struktur.

Kritiska framgångsfaktorer för struktur:

- En begriplig struktur
- Viktigt med övergripande struktur
- Ta ställning – detaljer eller summadata
- Denormalisering för slutanvändaren eftersom det är enklare att förstå

3.5.3 Process

Med process menas här två olika saker. Dels den process som det innebär att bygga upp ett datalager (speciellt viktiga delprocesser är exempelvis förankring, kvalitetskontroll, utbildning och förvaltning). Dels det sätt man beskriver och förhåller sig till de interna processer som datalagret ska stödja.

Kritiska framgångsfaktorer för process:

- Verksamhetsdriven process
- Utveckla stegvis
- Top-down eller bottom-up?
- Beställaren bör ha kvalitetsansvaret

3.5.4 Teknik

Vid datalagerdesign är det ofta tekniken som är den övervägande faktorn och ofta ägnas tekniken så mycket uppmärksamhet att de övriga faktorerna kommer i skymundan. Men odiskutabelt är att databasens, och de kringliggande verktygens (exempelvis flyttverktyg och preprocessning) egenskaper är viktiga för framgången. Enkelhet, kvalitet, kapacitet och säkerhet är viktiga frågor som man bör ta ställning till.

Kritiska framgångsfaktorer för teknik:

- Skalbarhet är viktigt
- Vem ska leverera hård- och mjukvara?
- Hur uppdateras datan, kontinuerligt eller grupperat?

3.5.5 Individ

Datalagret kommer alltid i slutändan att definieras av hur slutanvändaren använder det. Nyckelord för individen som användare är utbildning och användarstöd.

Kritiska framgångsfaktorer för individ:

- Slutanvändaren ställer kraven
- Slutanvändaren är knapptryckare utan mellanhänder
- Det måste finnas begriplig metadata
- Användarens miljö och stöd ska vara bekväm och begriplig

3.5.6 Organisation

Organisationen är det sammanhang som datalagret ska fungera i. Därför är det viktigt att lagret är anpassat efter organisationen. Detta gäller till exempel mål, både övergripande och detaljerade, terminologi och ansvar. Att tänka på:

- Förankra innan projektstart
- Utse tydligt förvaltaransvar
- Datalagret kräver stödorganisation

3.6 Sammanfattning

Business Intelligence hjälper alltså beslutsfattare att arbeta *med* mål och *mot* mål. För att implementera ett framgångsrikt BI-system krävs att vissa saker är uppfyllda. Det måste finnas mål som är specifika, mätbara, accepterade, realistiska och tidbestämda. Detta utgör basen för hur man sätter upp systemet. Nästa förutsättning är att det finns ett tekniskt källsystem som kan uppfylla befintliga krav på datalagring informationsinnehåll, kraven kan exempelvis identifieras med hjälp av de olika stegen i underrättelsecykeln. Detta är vanligtvis ett datalager. Utöver dessa två finns ett antal faktorer som är viktiga att beakta för att implementeringen ska lyckas. Dessa är; vision, struktur, process, teknik, individ och organisation. Allt detta bidrar till ett framgångsrikt och stödjande system.

4 Metod

Jag har i metodavsnittet haft för avsikt att resonera kring olika tillvägagångssätt och tankegångar under arbetet. Jag har delat upp rubrikerna tematiskt efter de olika huvudproblemen.

4.1 Om arbetet från problem till lösning

Jag ställdes inför ett problem som var abstrakt och konkret samtidigt. Problemet var framlagt på ett sådant sätt att det var greppbart och tydligt, eftersom jag förstod vad det var som förväntades av mig. Men samtidigt, hur utreder man ett behov, och vilken av alla möjliga riktningar är den mest lämpade? En stor del av arbetet har kommit att kretsa kring att göra rimliga avgränsningar, som diskuteras i ett separat stycke.

Jag började med att efter bästa förmåga försöka förstå uppgiften, och de olika begrepp, system och den organisation som låg till grund för problemet. När jag tyckte att bilden började klarna satte jag igång att i närmre detalj försöka utreda de behov som fanns och vilka problem som fanns med dagens situation. Detta ledde till en lista av problem med dagens lösningar, både praktiska problem och användarbehov som inte tillgodoses. Med problemen som bas, samt examensarbetets grundramar för ögonen, försökte jag tänka ut potentiella lösningar på problemen och formulera dessa i tydliga ordalag. Allt detta (utredningsfasen) låg till grund för den kravspecifikation som jag därefter sammanställde.

Efter att ha utrett problem och behov, och sammanställt ett kravdokument vidtog nästa fas, utvecklingsfasen. Jag försökte efter bästa förmåga att överföra mina tidigare resultat till en tänkt prototyp. Utvecklingen innefattade, grovt sett: laddning, modellering och design samt viss validering. Testning har varit en kontinuerlig process under arbetet. En del problem under arbetets gång har bestått i att på ett logiskt sätt försöka efterlikna tidigare rapporters (BEx/Excel) struktur i QlikView eftersom jag fått bygga upp datastrukturen från grunden. Hela kubstrukturen och alla nycklar har fått återskapas på nya sätt.

I och med att prototypen var avslutad var den praktiska delen av arbetet utförd. Den avslutande fasen, analysfasen, har innefattat reflektion, rekommendationer samt rapportarbete.

4.2 Perspektiv på teorin

Det teoretiska arbetet, som benämns *Att arbeta med Business Intelligence*, ska ses som ett grundläggande ramverk som fyller tre syften. Det beskriver grundläggande begrepp som är viktiga för *läsarens* förståelse, det har naturligtvis också bidragit till att *jag* lyckats förstå olika komponenter för att kunna programmera och ladda (till exempel begrepp som multidimensionella kuber är helt centrala för att förstå hur man ska gå till väga) och teoriavsnittet ger även en grund för analys. Arbetet har övergripande innefattat två analysfaser, en inför prototypframställningen, och en reflektion efteråt. Det teoretiska ramverket används främst för analys innan prototyparbetet, den avslutande analysen innefattar personliga åsikter kring användande, fördelar, nackdelar etcetera.

Som en kommentar till den litteratur jag använt mig av kan man säga att området kring Business Intelligence är något som snarare praktiseras än teoretiseras. Eftersom området är starkt växande så förändras applikationer och möjligheter snabbare än litteraturen hinner med. Detta innebär att de förhållandevis få böcker som finns om området ofta är föråldrade. Det räcker med några år så har verkligheten förändrats totalt. Jag har inte heller hittat många vetenskapliga artiklar inom ämnet. Detta är något jag haft i åtanke då jag beaktat de tillgängliga källorna.

4.3 Att utreda behov och användare – intervjuförfarande

En inventering av lämpliga respondenter gjorde genom en inledande kartläggning. Jag ville kontakta

- Användare av nuvarande system
- Skaparna av de nuvarande rapporterna
- Skaparna av det nya tidsuppldelningssystemet
- Potentiella användare av ett nytt system
- Expert på den aktuella programvaran

Utifrån mina prioriterade kategorier har jag genomfört ett antal intervjuer med personal på SSD. Riktlinjer för intervjuerna har de flesta fallen varit följande. Jag har sammanställt en intervjumall (se bilaga 7) med övergripande frågor. Respondenterna har kontaktats och fått veta på ett övergripande plan vad jag är intresserad av att prata med dem om. På överenskommen tid och plats har jag sedan hållit en semistrukturerad intervju med stöd från intervjumallen. Tanken med förfarandet är att ha ett strukturerat tillvägagångssätt för att inte missa viktiga ämnesområden, men så löst så att det ändå funnits utrymme för respondenterna själva att ta upp egna punkter eller diskussionsfrågor. Jag har fört anteckningar under intervjuerna och gjort en skriftlig sammanställning direkt efter avslutad intervju. De intervjuer som inte följt dessa riktlinjer är de personer som jag haft kontinuerlig kontakt med under arbetets gång, exempelvis min handledare och min QlikView-expert, samt vid vissa tillfällen även min beställande chef.

4.4 Avgränsningar

Arbetet har på många sätt handlat om att göra rimliga avgränsningar, främst under prototyparbetet. En del avgränsningar har handlat om övergripande frågor, exempelvis vad som ligger inom arbetsuppgiften, medan en stor mängd har varit på detaljnivå gällande olika val, modeller, vad som ska tas med eller ej, hur lika modellen och summeringarna måste vara originalet samt frågor gällande datamängderna. Jag ska försöka sammanfatta de viktigast avgränsningar som gjorts.

En grundläggande avgränsning gjordes i inledningsskedet av arbetet, där uppgiften kring prototypbygget specificerades till QlikView. Detta gjordes av flera anledningar, dels för att utredningsarbete kring programvaran med stor sannolikhet hade omöjliggjort ett prototypbygge, på grund av tidsbrist och dels är den valda programvaran något som det fanns kompetens inom på företaget, eftersom det är en tjänst som erbjuds externt till kund, och det var därför lämpligt att testa programmet även inom intern uppföljning. Jag behövde

inte heller lösa driftsfrågor eller uppdateringar i prototypen, dock gärna tänka på möjliga lösningsförslag.

Resten av avgränsningarna handlar om prototypen. Maj 2007 valdes som exempelmonad av rent praktiska skäl, det var i juni som laddningarna skedde, och tidigare månads data utgjorde en naturlig startpunkt.

I den Excel-rapport som redan fanns gjordes ett antal specialkorrigeringar för att "rätta till" felaktigheter i dataladdningar. Jag har hanterat dessa i möjligaste mån men inte helt. Det jag hanterat är att filtrera bort specifika teamkoder, samt vissa personer som betraktas som over head, alltså icke budgeterade för. Det jag inte har hanterat är hur resurser från Gimo (där ca 5 personer från SSD arbetar) hanteras, samt den sjukdomstid som läggs in från IFS-systemet manuellt.

Jag har även löst specifika problem längs vägen som kan betraktas som avgränsningar. Ett problem har legat i hur nycklarna (mellan tabeller) ska utformas i modellen för att bäst matcha verklighet, samt de övriga rapporterna, och där har jag fått ta ställning i varje enskilt fall vilken som är den rimligaste lösningen. De tre projektnivåerna var en annan nöt att knäcka eftersom de har legat i varsin separat tabell i BW, men för att QlikView-kopplingarna skulle fungera behövdes en annan lösning. Min QV-expert hjälpte mig med att konkatenera (datorterm för att sätta ihop, göra en en s.k. *join*) de olika tabellerna till en stor.

Att föra in budgetjämförelse som en funktion var en avgränsning så till vida att jag endast läste in de data som gällde för maj, och endast för de tidskategorier jag hade tillgång till budget för. De kategorierna som fanns budgeterade var den gula och den blå tiden.

4.5 Sekretess

SSD har låtit mig publicera material kring deras system och datorarkitektur. Prototypen i sin helhet, koden och data beträffande SSD utgör inte en del av rapporten.

5 Beskrivning av nuläge

För att få en bra grundförståelse för vidare analyser har jag försökt sammanfatta dagsläget kring rapporteringen.

5.1 Bakgrund

Personalen på SSD rapporterar sin arbetstid på två sätt; på "stämpelklocka"(numera ett magnetkort) vid ankomst och hemgång. Dessa tider registreras i programmet IFS och skickas sedan vidare till Sandviks tidsrapporteringssystem Projus. I Projus rapporterar medarbetarna sin tid som lagts timvis på olika projekt.

För att få en bra bild över hur personalen lägger ned timmar och därigenom hur företaget går ekonomiskt har SSD:s ledningsgrupp tagit fram ett kategorisystem för att se på den rapporterade tiden (beskrivs mer utförligt nedan). Utifrån denna uppdelning har det tagits fram Excel-ark med de efterfrågade timmarna, uppdelade efter olika kriterier.

5.2 Hur produceras rapporterna idag?

Jag har försökt klarlägga hur rapporterna produceras idag. Detta är inte helt enkelt. Det finns nämligen olika versioner, som används en aning olika.

Det finns en rapportform som ligger inbyggd i SAP-BW, som är byggd för att lyfta fram en del av den information som finns där. Det görs via det verktyg som heter Business Explorer, kallad BEx-rapporter. Det finns två relaterade BEx-rapporter som är aktuella i detta fall. Den ena visar total rapporterad arbetstid mot de timmar som rapporterats specifikt, denna rapport heter "Scorecard IFS". Den andra visar hur tiden rapporterats på olika projekt och heter "Project Follow Up". Dessa rapporter genereras från programmet självt efter att relevant data laddats in, efter att de en gång skapats och alla kategorier har definierats. Tyvärr är rapporterna så svårhanterade att ingen som skulle ha nytta av dem faktiskt tittar på dem. De efterfrågade siffrorna ligger i ett Excel-dokument som är resultatet av en utsökning (en query) från informationskuben. Utsökningen kan man modifiera via användargränssnittet, men det kräver tillgång till samt viss vana av Business Explorer.

Eftersom de nämnda rapporterna inte används, då de är krångliga att komma åt och svårmodifierade, samt att man faktiskt har ett behov av den information som datalagret innehåller, har man genomfört en alternativ lösning. Genom en utredning tog man fram de data som man önskade att rapporten skulle innehålla (bilagan är borttagen ur denna versionen). Dessa önskemål gick sedan vidare till SSD:s ekonomiavdelning (benämnd DAF). De har utifrån önskemålen skapat ett Excel-dokument med den önskade informationen. Detta Excelblad består av 17 flikar som hämtar data från de tidigare nämnda BEx-rapporterna. Datan är omstrukturerad efter användarnas önskemål, vilket innebär att den är grupperad på ett annat sätt än BEx-rapporterna. Det finns även andra detaljändringar, i form av ett antal undantag som måste läggas in "för hand". Detta sker genom att man varje månad måste gå in manuellt och hämta ut önskad information och sedan justera för vissa tillfälliga förändringar. Detta kan gälla att ta bort eller lägga till tid, eller ändra tid från extern till intern tid eller modifiera s.k. over head-tid. Enligt en grov uppskattning tar arbetet en dryg arbetsdag per månad att sammanställa, utspjutt över flera dagar. Ett problem

med denna rapportform är att arbetet inte kan påbörjas förrän efter den 10:e var månad eftersom tidsstämplingen ska attesteras av närmsta chef, vilket innebär en fördröjning för dataflyttningen.

En tredje rapportform är den Excelrapport som hämtas direkt från Projus, tidsrapporteringsystemet. Detta är en lösning på att man varit tvungen att vänta till efter den 10:e för att få säkra siffror, vilket beror på att rapporterade siffror ska godkännas av en högre instans innan de kan laddas, vilket gör att det drar ut på tiden. I denna tredje rapport hämtas datan direkt från Projus, uppdelad på varje kompetensteam, och kan göras när som helst i månaden. Detta för att få en snabb överblick över dagsläget, men är egentligen inget verktyg man kan bygga hållbara prognoser på.

5.3 Vem använder rapporterna?

Beställaren för den aktuella studien är SSD:s ledningsgrupp. De vill se över de rapportformer som finns att tillgå i dagsläget.

Excelrapporten som DAF producerar är till främst för ledningsgruppen, som använder dem för att bedöma hur företagets resurser används.

Bex-rapporterna verkar ingen använda, utom att de ligger till grund för ovan nämna Excelrapport.

De rapporter som tas direkt från Projus används i dagsläget av teamcheferna som ögonblicksbild av hur deras team rapporterat olika tider, och kan i bästa fall leda till en proaktiv styrning. Men jag ser även en potential att teamchefer och övriga med leveransansvar kan ha nytta av rapporterna i sitt planeringsarbete.

5.4 Struktur och nomenklatur

Den rapporterade tiden är uppdelad i olika kategorier, grön, blå och gul tid⁴. De olika kategorierna markerar olika typer av aktiviteter personalen lägger sin tid på.

Grön tid markerar den debiterbara tiden mot kund, alltså olika fasta projekt och avtal. Det kan vara licenser, fastprisavtal eller andra externa avtal. Blå tid innefattar den interna produktiva tiden, såsom interna förbättringsprojekt, kvalitetsarbete osv. Gul tid är övrig tid, som innefattar försäljning, utbildning, interna möten och aktiviteter och idle.

Resurser är den personal som är tillgänglig, varje person är en resurs (en resurs kan teoretiskt sett också vara exempelvis en maskin eller dator som utför arbete). De resurser som är markerade som OH är så kallad *over head* vilket innebär att de inte är budgeterade att producera något i projekt, dessa personer är oftast chefer eller eventuellt administrativ personal. All eventuell rapporterad tid från dem räknas bonus för företaget, alltså ren vinst som inte budgeterats för.

⁴ I applikationen kallas dessa: grön-External debet, blå-Internal projects&assignments(P&A) gul- Other activities

Resurserna jobbar oftast i projektform. Ett projekt kan ha tre projektnivåer registrerade i Projus (vilka sedan följer med genom systemen). Nivåerna är hierarkiska, där det alltid finns en projektnivå 1, och sedan kan det finnas nivå 2 och nivå 3 kopplade till toppnivån om projektet ha flera delgrenar. Projekten har även andra kategorier som exempelvis en projektledare, team code, beräknat slutdatum etcetera⁵.

Gällande de olika tidstyperna i systemet så finns det tre olika typer av rapporterad tid. *IFS-tid* är den tid som kommer direkt från tidsstämplingssystemet med samma namn. *Follow Up-tid* är de timmar som medarbetarna lägger ned på varje projekt, varje dag. *Scorecard-tid* är en summering av Follow Up-tiden på en dag.

Jag illustrerar med ett exempel. Om du jobbar på SSD och kommer en dag 8.00, stämplar ut för lunch en timme och går hem 17.00 samt jobbar parallellt med två projekt, Projekt A två timmar på förmiddagen och Projekt B sex timmar resten av dagen kommer tiden att se ut som följer:

IFS-tid: $8.00-17.00 = 9h - 1h \text{ lunch} = 8h$

Follow Up-tid: Projekt A 2h + Projekt B 6 h

Scorecard-tid: $2h + 6h = 8h$

Just i exemplet stämde alla tider väldigt väl, men i verkligheten är det väldigt sällan så att tiderna stämmer överrens. Detta är en av anledningarna till att man vill undersöka vad tiden egentligen går åt till.

⁵ För detaljer hänvisar jag till bifogade tabellstrukturer

	Internal	2007-05
	Internal meeting	734
	Recruitment	91
	Team meeting	508
	Personal: Objective	255
	Guidance of new empl	268
	Feedback delivery	129
InternalTotal	Total	1 984
InternalBudget	Budget	1 411
InternalTotal, acc	Total, acc	7 946
InternalBudget, acc	Budget, acc	7 055
	 Ovrig Gul tid	
	Grand total	
Grand totalGrand Total	Grand Total	5 193
Grand totalBudget	Budget	3 828
Grand totalGrand Total, acc	Grand Total, acc	19 565
Grand totalBudget, acc	Budget, acc	19 141

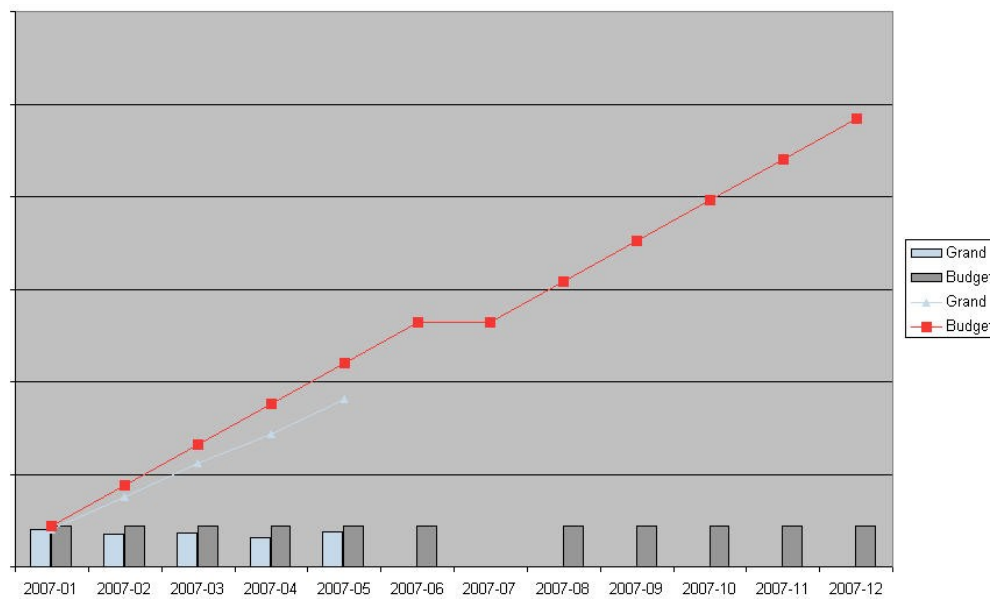
Figur 7 - Intern tid och sammanställning av gul tid i Excel-rapporten

I flikraden i figur 6 ses *Sammanställning SSD*. Där sammanställs siffror från de olika flikarna för att ge en helhetsbild. Där presenteras bland annat total tid, de olika tidskategorierna, IFS-tid och ackumulerad tidssummeringar.

Period	2007-05
IFS Available hours	30 756
IFS Sickness	197
External debet hours	20 620
External other	3 912
Internal P & A	1 863
Internal hours	5 193
External debet qouta %	79,3%
Diff IFS Vs Projus reported hours	832
Diff IFS Vs Projus reported hours, acc	-816
Accumulated	2007-05
IFS Available hours, acc	145 689
IFS Sickness, acc	2 953
External debet hours, acc	94 575
External other, acc	21 706
Internal P & A, acc	9 027
Internal hours, acc	19 565
External debet qouta %, acc	78,2%
Budgeterade timmar externt	233 23 534
External debet hours, OH	221

Figur 8 - Sammanställning i Excel-rapport

Utöver dessa tabellsummeringar finns det också tre stycken diagram i rapporten, en för varje tidskategori. Dessa visar hur det går månad för månad för en definierad kategori, se exemplet nedan.

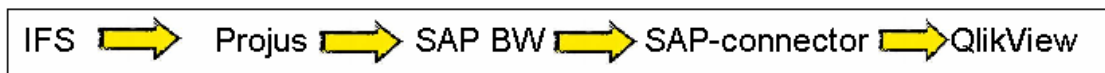


Figur 9 - Diagram i Excel-rapport

Den Excel-rapport som gör med data direkt från Projus är snarlik den ovan beskrivna rapporten men är inte lika detaljerad. Där finns enbart två flikar, en för sammanställning av data (görs ett per team) och en flik för ett diagram liknande det ovan.

6 Översikt av programvara

I detta avsnitt går jag igenom de olika programvaror som är aktuellt för arbetet. Jag beskriver dessa i övergripande termer, med fokus på den funktionella rollen, inte på den tekniska prestandan. De två programmen som har använts mest är QlikView och SAP BW, varför dessa är mer detaljerat beskrivna. I skissen nedan ses en schematisk bild av hur laddningarna går till, steg för steg.



Figur 10 – Schematisk skiss över dataladdningarna

6.1 IFS

IFS (Industrial and Financial Systems) är Sandvik System Developments bastidsrapporteringsystem. Det är detta system som loggar ankomst och hemgång via en terminal i entrén där du drar ditt passerkort. Det är också via detta system som du manuellt kan rapportera in arbetade timmar då du inte dragit kortet (på resa, jobbat hemifrån etc). Systemet är fristående från övriga system och håller enbart reda på antal arbetade timmar, men inte vad de spenderades på.

6.2 Projus

I Projus däremot går varje anställd in och rapporterar hur de spenderat tiden på jobbet. Eftersom SSD är ett konsultbolag (med bara Sandvik som kund) spenderar medarbetarna sina timmar på olika sätt, vilka debiteras olika, och detta måste det finnas kontroll för. Dessa inrapporterade timmar lagras i en datastruktur, där den ligger till grund för företagets fakturering. Månadsvis tankas informationen över till SAP:s datalager, kallat Business Warehouse.

6.3 SAP Business Information Warehouse (SAP BW)

SAP är en av världens största leverantörer av affärssystem. BW-applikationen är deras Data Warehousing-lösning som används av Sandvik Systems Development. Strukturen är uppbyggd av tre principiella funktioner:

6.3.1 Laddning av data till datalagret

Detta innefattar även extraktion och eventuell transformation som krävs för att få den önskade strukturen på lagret. I det aktuella fallet handlar det alltså om laddning från IFS och Projus till SAP BW, vilket vanligtvis görs en gång i månaden, med undantag för vissa data som laddas veckovis.

6.3.2 Datalagret

I datalagret sparas den transformerade datan i så kallade informationskuber. Begreppet kub används för att visa att datastrukturen är flerdimensionell. Det innebär i vårt aktuella fall att

det finns en faktatabell (tidsrapporteringsdata) och från denna utgår ett antal dimensioner (likt en stjärnstruktur, se teoriavsnitt om datadimensioner).

6.3.3 Rapportering

Verktyget för den funktionella analysen kallas Business Explorer, förkortas BEx, och det är detta verktyg man använder idag på SSD. Verktyget hjälper till att producera rapporter i Excel-format med vald data.

6.4 SAP-connector

För att kunna ladda in data från SAP BW till QlikView krävs ett så kallat middleware för att det ska fungera. Till detta används SAP Connector för att plocka ut data ur SAP BW-kuben och ladda över till QlikView för användning.

6.5 QlikView

QlikView görs av Qliktech International AB som har sitt säte i Lund och försäljningskontor ett stort antal försäljningskontor världen över. QlikView är en modern OLAP-applikation med stöd för ett grafiskt användargränssnitt. En hel analyskub komprimeras och laddas i RAM-minnet. Gränsen för hur stora tabeller och fält får vara är i det närmsta obegränsad så datorns internminne sätter gränserna för hur stora datamängder som kan behandlas.

Eftersom varje post finns snabbt tillgänglig görs varje beräkning först när den efterfrågas. Den önskade informationen går då mycket fort att då fram enbart genom ett antal klick, därav namnet – klicka och titta.

I en QV-applikation modelleras det mesta via ett script. Genom scriptet laddar man in datamängderna och formar sedan de tabeller och nycklar man är intresserad av. Man kan även göra en mängd andra saker som att byta namn, exkludera mängder, konkatenera tabeller mm. Scriptspråket är enkel och grundläggande programmeringssyntax. De data man laddar in via scriptet blir sedan tillgänglig via ett grafiskt gränssnitt där all presentation sker. Gränssnittet skapas inte via scriptet.

Jag har samlat lite mer ingående information om hur jag laddat in data från BW till QV och lagt som en bilaga för nyfikna läsare.

7 Förutsättningar för en ny lösning

Mycket av materialet detta avsnitt är hämtat från mina intervjuer, men jag har satt in det i olika analytiska sammanhang, i ett kravdokument eller i modellen för kritiska framgångsfaktorer.

7.1 Stjärnstrukturer i praktiken

De data som finns lagrat i BW-kuberna har inte en alldeles enkel struktur. Här tänkte jag redogöra för ungefär hur det är tänkt att förstås. Som förståelsehjälp vill jag referera till de tabellstrukturer som ligger som en bilaga till arbetet (bilagan är borttagen i denna version).

I BW finns tre faktatabeller, en för varje tid som lagras, alltså IFS-tid, Scorecard-tid och Follow Up-tid. Faktatabellerna (alltså transaktionsraderna) är utbyggda med ett antal dimensionstabeller. Några av dessa dimensionstabeller är:

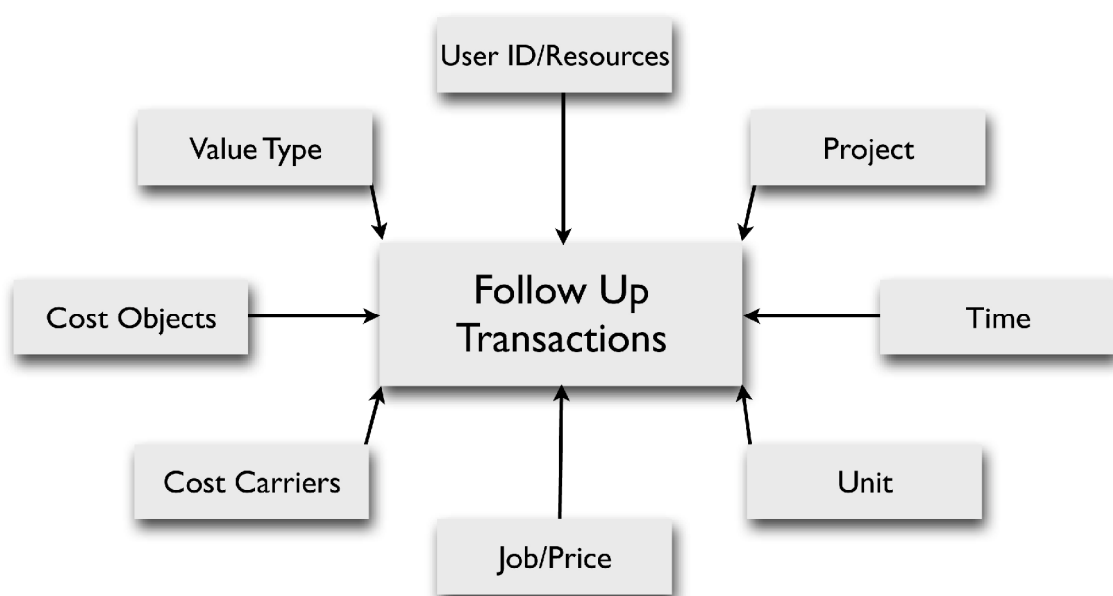
Resurser – innefattar all information om de resurser (anställda) som finns registrerade

Projektinformation – all information om projekten finns i dessa tabeller, en för varje projektnivå, totalt tre stycken

Priskoder – information om vad de interna konsulterna med olika kompetens kostar

Jobbkoder – information om vilka olika yrkeskategorier som finns

För att visa hur strukturen ser ut har jag infogat stjärnstrukturen för den mest komplexa kuben (den transaktion med flest dimensioner). Se Follow Up-tid nedan, figur 10.



Figur 11 – Stjärnstruktur för Follow Up-tid

I alla dimensioner i figuren finns ett antal olika fält som jag har uteslutit för att förbättra förståelsen. För detaljerade strukturer med alla fältnamn för alla tre tidstyper se se bilaga 6 (bilagan är borttagen i denna version).

7.2 Underrättelsecykeln i praktiken

Av det arbete som beskrivs i modellen för hur information bör analyseras i underrättelsecykeln är det endast en del som är aktuellt inom detta arbetes ramar. Detta är ett försök att placera in informationsbearbetningen i sitt sammanhang just gällande denna BI-lösning.

Beslut om hur man ska inrikta data har redan gjorts, på två sätt. Dels har strukturen för Projus och BW bestämts, och genom dessa design- och funktionsval har inriktningen begränsats. Det andra beslutet som inriktat informationen är de beslut man tagit gällande strukturen på tidskategorierna som definierats. Dessa två beslut tillsammans bildar en mall för hur man bör inrikta informationen.

Inhämtningen sker genom att personal registrerar tid på olika sätt, både via kort och via Projus. I dagens läge skickas även ett fåtal tidsrapporter via manuella Excelark.

Det är bearbetningen som varit fokus för denna utredning, samt i viss mån delgivningen, och på vissa sätt kan man säga att arbetet skett iterativt. Jag har undersökt vilka delgivningsbehov som funnits, och försökt applicera de resultaten på bearbetningen, som jag gjort från gräsrotsnivå i form av ett nytt applikationsförslag.

7.3 Förundersökning

Jag gjorde en förundersökning i kombination med nulägesanalysen där jag genom intervjuer försökte få fram de konkreta problem som fanns med dagens lösningar. Dessa problem försökte jag sedan finna lösningsförslag på i och med den tänkta QlikView-applikationen. Resultatet sammanfattar jag nedan i punktform för att vara så tydlig som möjligt.

7.3.1 Problem med dagens rapporter

1. Olika personer har olika siffror eftersom det finns olika rapporter
2. Rapporterna tar lång tid att sammanställa, vilket är kostnadskrävande
3. Kunskapen är låst till en person
4. Rapporterna är svåröverskådliga
5. Alla har inte möjlighet att se just de som de vill, utan får nöja sig med förutbestämda val
6. Man kan inte drilla i informationen, den är statisk och mycket oflexibel

7.3.2 Lösningar på dessa problem

1. Bara en rapport!
2. Gör detta automatiskt så långt det är möjligt, och spara på så sätt tid och pengar

3. Om lösningen programmeras i en applikation, är det enklare att föra kunskapen vidare, och det kräver inte lika mycket av en administratör
4. I QV byggs rapporterna upp på ett mycket mer lättöverskådligt sätt, eftersom man slipper faktabladen (alltså det bakomliggande lagret)
5. Poängen med QV är just att man ska kunna göra alla val man vill
6. I QV är informationen dynamisk, att drilla går lätt

7.4 Kravspecifikation

Utifrån de specificerade problemen och möjliga lösningar sammanställde jag en kravspecifikation av en teknisk lösning som var tänkt att tillgodose behoven.

Kravspecifikationen i sin helhet finns bifogad som en bilaga, men ger här en överblick av vad den innehåller.

7.4.1 Funktionell beskrivning

Under denna rubrik i kravspecifikationen har jag försökt att i så konkreta ordalag som möjligt specificera de tekniska och funktionella krav och sammanhang som omger systemet. Alltså hur det rent konkret borde se ut och vad man ska kunna göra.

7.4.2 Konsekvenser

De olika områden som jag uppfattar att systemet kommer att få konsekvenser inom tar jag upp var för sig. Dessa områden är; tillgänglighet, påverkan på befintligt rapportsystem, verksamhet, drift, utbildning samt migrering av data.

7.4.3 Tolkning av användarkrav

Jag ger även en kravspecificering gällande de användarkrav som uppkommit under utredningsarbetet. Jag har försökt att, så konkret som möjligt, beskriva hur dessa krav kan tolkas praktiskt, och därmed även hur problemen kan lösas.

7.5 Kritiska framgångsfaktorer i praktiken

Arbetet har inte varit att ta fram en organisations datalager, utan en OLAP-applikation baserat på ett datalager. Jag uppfattar dock problematiken som snarlik och det är min åsikt att de framgångsfaktorer som Söderström tar upp också kan appliceras på fall. De saker som tas upp i texten refererar till de faktorer som jag redogjort för under motsvarande teoriavsnitt.

7.5.1 Vision

Hur man bör jobba med prioriterade mål och strategier tar jag upp under i de delar av arbetet som berör målarbetet (SMART-modellen samt målarbetet i praktiken).

Det är viktigt att uppmärksamma de vinster som görs. Det kunde vara av stort värde att uppmärksamma något fall där rapportapplikationen varit till hjälp i en fråga.

Det är även viktigt att göra ledningsgruppen samt övriga potentiella användare uppmärksamma på nyttan och möjligheten med systemet.

7.5.2 Struktur

Då man arbetade fram det nya tidskategorisystemet var målet att få en begriplig struktur över det som man ville analysera. Detta är något jag försökt föra vidare till prototypen, och varit så trogen den strukturen som det gått att vara. Fördelen med tidskategorierna man satt upp är att de är ganska övergripande, men detta kan också vara (till exempel programmeringsmässigt) en nackdel eftersom det skulle kunna uppkomma gråzoner.

I en QlikView-applikation finns det en stor fördel i att man inte behöver välja mellan summadata (summerade beräkningar) eller detaljer, det är enkelt att göra hopsummeringar, samt ändå kunna få se datan på detaljnivå vid intresse. Datastrukturen i QV är denormaliserad.

7.5.3 Process

Det är naturligtvis målsättningen att arbetet med att ta fram en ny applikation ska vara verksamhetsdriven. Jag har försökt så långt det varit möjligt att jobba med verksamheten för ögonen, men vill påpeka att det funnits vissa praktiska svårigheter med detta då jag inte varit så insatt, eller inblandad i den faktiska verksamheten. Detta blir därför en rekommendation till vidare arbete, att i möjligaste mån utgå ifrån verksamhetens krav och processer. Resonemanget går hand i hand med att processen i applikationen bör vara tillämpningsdriven.

Applikationsarbetet har utvecklats stegvis, men kanske inte i så iterativt samarbete med användarna som jag hoppats på. Man kan se prototypen som ett första steg, där ytterligare arbete kan bygga vidare på den kravspecifikation och de lärdomar som dragits. Top-down eller bottom-up lösning är ett designval som inte behövts. Eftersom de stora datamängderna laddas över i sin helhet och sedan modelleras efter en färdig struktur så kan man säga att det varit både top-down och bottom-up var för sig, med kanske en aning fokus på det senare.

Att beställaren har kvalitetsansvaret är något som kan vara viktigt att poängtera. Mycket teori kring användbarhet tar upp just beställarens ansvar, och pratar om att beställarkompetens är nödvändigt för att få ett väl fungerande system. Detta är inte något som varit en del av mitt arbete, men något som jag kommer att rekommendera vid ett eventuellt vidare arbete mot en lösning.

7.5.4 Teknik

Gällande tekniken så menar Söderström att skalbarheten är viktig, att man bör kolla upp leverantör samt bestämma uppdateringsregler. Jag har enbart skapat en prototyp som gäller för en månad. Men, utan att ha testat, så tror jag att skalbarheten att få upp datamängden till en tidsperiod av 2-3 år inte vore problematiskt. Laddningstiden är i dagsläget runt 30 sekunder, men vid större laddningar är det i stort sett bara faktatabellerna som kommer att öka i omfång så laddningstiderna borde vara hanterliga. Detta behöver bara göras på månadsbasis, och kan exempelvis göras natttid.

Frågan om vem som levererar komponenter är redan avgjord i och med att avgränsning för arbetet gjordes, prototypen skulle utvecklas i QlikView som levereras av QlikTech. Datamängden i applikationen kan uppdateras på tre sätt. Antingen låter man laddningen vara tidsbestämd, exempelvis att den laddas den första dagen på varje månad. Ett annat alternativ är att laddningen triggas av en händelse. I detta fallet skulle det till exempel vara att QlikView uppdateras då det laddats in nya tidsdata från Projus till BW. Detta görs på månadsbasis. Ett sista alternativ är att det sköts manuellt, att en person måste gå in och ladda det vid behov, vilket jag inte rekommenderar.

7.5.5 Individ

Jag har samlat ihop krav både från slutanvändarna av dagens rapporter och tänkta nya användare. Min ambition vid utveckling har varit att tillgodose de kraven. Det är min rekommendation att fortsätta arbeta på det sättet vid en eventuell vidareutveckling.

Metadata finns på olika sätt. Tidsstrukturen beskriver de olika tidstyperna, datastrukturen i QlikView visas i tabellschemat (i kapitel 8.1.1, figur 10). Det finns en enkel användarmanual under fliken *How To* i applikationen. Ambitionen är också att designen ska vara bekväm att använda och lättbegriplig.

7.5.6 Organisation

Om det ska införas ett nytt, skarpt, rapportsystem i QlikView så är denna sista kvalitetsfaktor av yttersta vikt. Ett system där medarbetarnas rapporterade tid läggs in, och besluten kring vem som ska ha tillgång till det samt hur det ska användas måste förankras. Även under mitt relativt korta arbete har jag märkt av hur stora delar av organisationen som är berörd. Det märks inte minns på bredden av respondenter. Det finns en mängd åsikter i frågan, och för att lyckas med systemet så är förankring helt centralt.

För att lösningen ska fungera rent praktiskt krävs även att ett tydligt förvaltaransvar utses. En stödorganisation som Söderström kallar det är kanske ett starkt ord för en enda intern BI-applikation, men innebörden är viktig. De som jobbar med de andra systemen (Projus och BW) bör kännas till applikationen, dess innehåll och vad som skulle hända om de fick för sig att ändra strukturen i de andra systemen, eller åtminstone att något kunde hända i tredje led. De eller den som utses till förvaltaransvarig bör vara medveten om sitt ansvar samt ha kunskap att genomföra specifika förändringar.

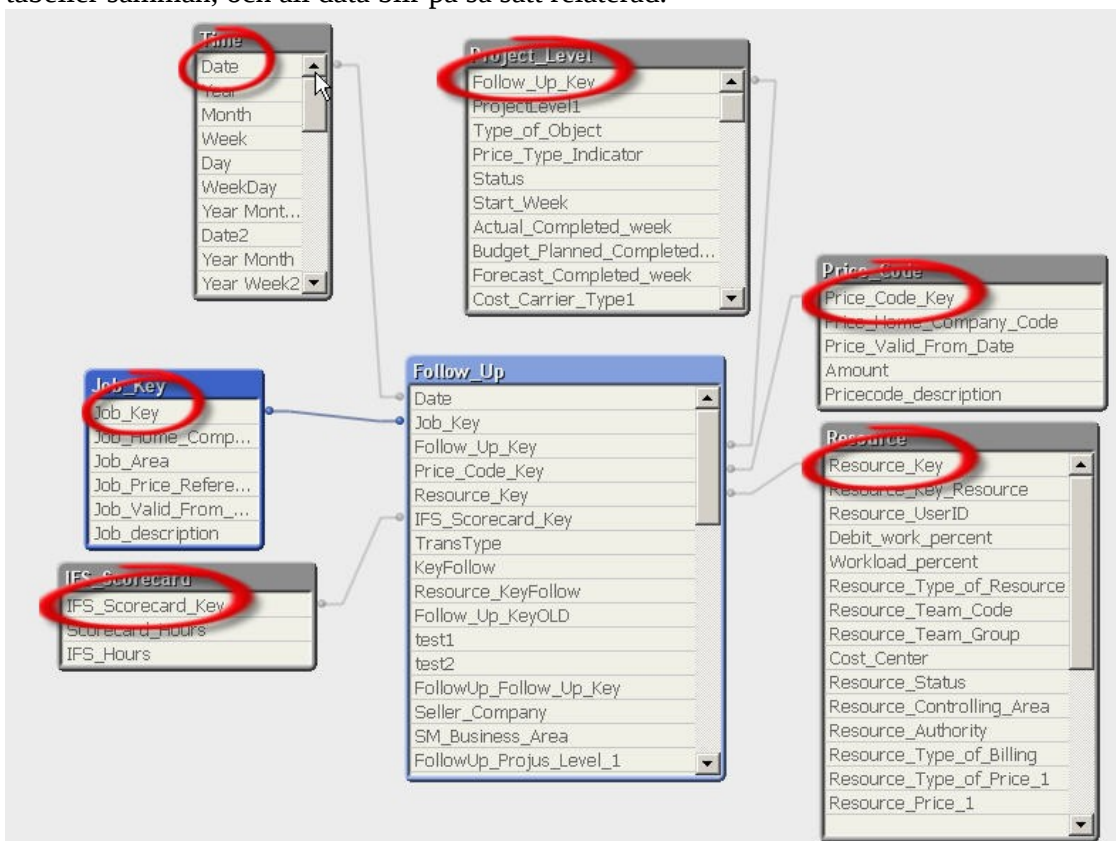
8 Prototypen

Att utveckla prototypen var ett handfast sätt att konfronteras med de problem som utretts under den föregående fasen av arbetet. Det är svårt att beskriva ett så interaktivt program som detta är i enbart ord och bild. Jag ska med hjälp av skärmdumpar försöka göra en övergripande beskrivning, koden finns bifogad för den tekniskt intresserade läsaren.

8.1 Design och innehåll

8.1.1 Datakopplingar

Kopplingarna som finns mellan de olika tabellerna görs genom att man definierar likadana nycklar i två tabeller (dessa skapas i scriptkoden). Genom att skapa nycklar kopplas olika tabeller samman, och all data blir på så sätt relaterad.



Figur 12 - Tabellkopplingar i QlikView.

8.1.2 Layout

Applikationen består av sex stycken funktionella flikar varav en instruktionsflik. De sex flikarna svarar mot varsitt så kallat blad (eng. sheet).



Figur 13 - Flikraden i QlikView

Budget består av en mätare av hur olika kategorier gått jämfört med budget. Resultatet finns även i tabellform.

Label

Follow up hours

Definition

Sum (Follow_Up_Hours*Green_Time_Flag)

Blå tid (*Internal P&A*) och gul tid (*Other activities*) fungerar som fliken för grön tid med det finns ingen debiteringsgrad på dess flikar.

To make selections

Selected Data (Green)

Associated Data (White)

Unrelated Data (Gray)

Locked Data (Blue)

To clear selections

Customer

Country	Cust. no.
Germany	3540-1021640
Name	Envirotec Ges.

Clear selection in active list box

To search for data

The figure consists of two screenshots of the Microsoft Access interface. The top screenshot shows the 'Customer' table with the 'Country' field set to 'France'. The 'Searching for' box contains the text 'fr*'. The bottom screenshot shows the same 'Customer' table, but the 'Country' field is now set to 'South Africa'. The 'Searching for' box now contains the text '*fr*'. Both screenshots show the 'Cust. nu...' and 'Name' fields as well.

To view current selections

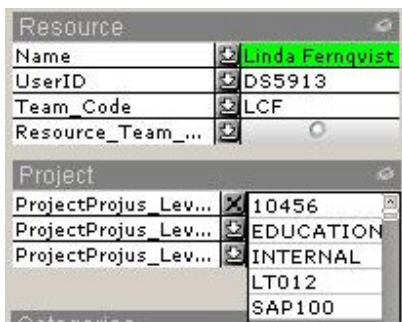
CURRENT SELECTION	
ES_SALESMANNAME	Birger Sjödin
S_SABCOUNTRY	SE
S_GROUPREGION	EMEA

36

8.1.3 Att göra sökningar

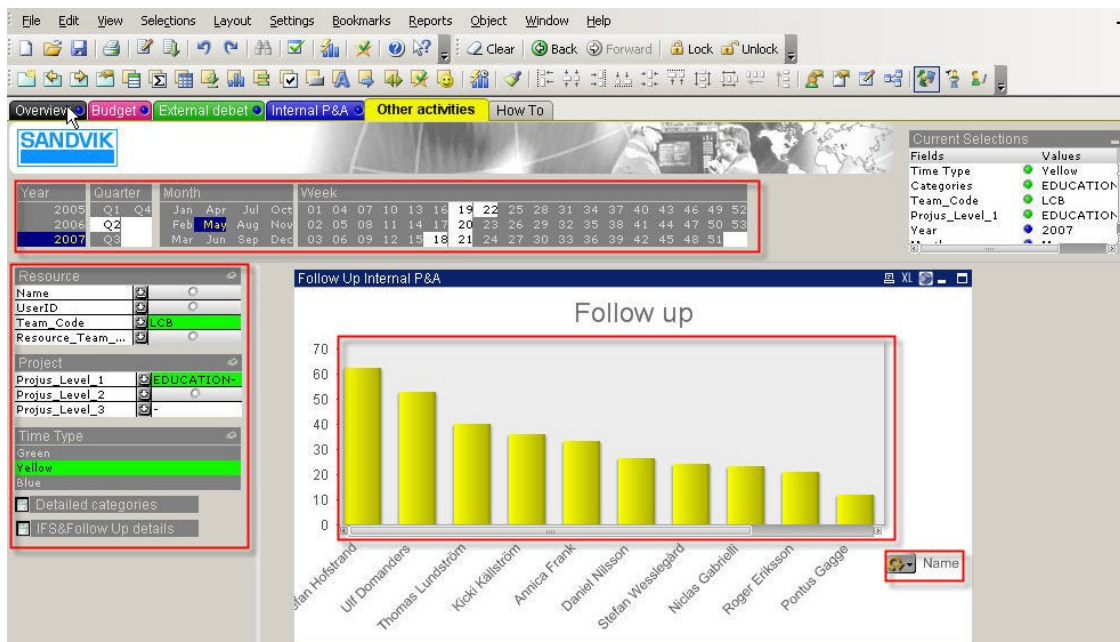
När man vill söka ut något i applikationen väljer på tre olika sätt. Dels väljer man vilken tidsperiod man vill titta på. Detta väljs i den översta delen av fönstret.

Man väljer även vilka utsökningskriterier som finns, vill man exempel se en enskild anställd eller ett specifikt team så markeras detta i vänstermenyn i de olika flikarna. I nedanstående exempel har jag valt resurs Linda Fernqvist. Om jag vill specificera ytterligare, vidare på projektnivå kan jag välja mellan de fem projekt som hon lagt tid på under den valda tidsperioden.



Figur 16 - Närbild på vänstermenyn i QlikView-rapporten

Utöver detta så väljer man också vilken visningstyp man är intresserad av, tabell eller diagramtyp samt vilken av storheterna som ska presenteras. Jag har markerat de olika valen i bilden nedan. I detta exempel är maj 2007 valt överst. I vänsterspalten är valet att titta på team LCB och den tid de lagt på utbildning. Presentationsval som gjorts är att vis i tabellform, med namn som presenterad storhet.



Figur 17 – Valmöjligheter i QlikView-rapporten

Man kan ta bort gjorda val genom att trycka "Clear" uppe i menyraden eller ta bort enskilda val genom snabbkommandot i övre högra hörnet på valrutorna i vänstermenyn. Det är även möjligt att låsa val, att göra vissa val till bokmärken som det är lätt att återkomma till, eller att backa tillbaks till de val man tidigare gjort.

8.2 Funktioner

De funktioner som finns i prototypen har jag utformat efter uppdragsgivare och andra användares önskemål, på basis av den förundersökning som föregick prototyparbetet.

8.2.1 Tidskategorier

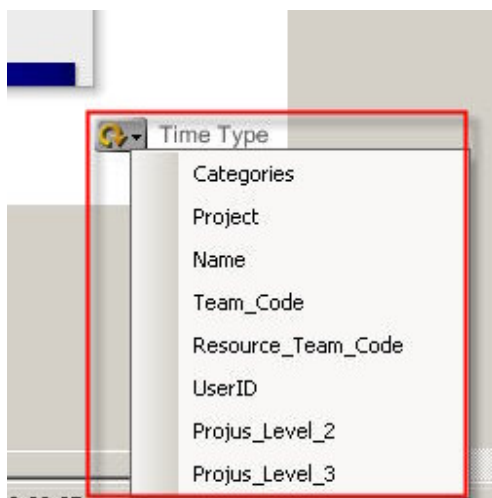
Alla klassificeringar i applikationen är gjorda utifrån klassificeringen av de olika tidskategorierna. Alla projekt är uppdelade i kategorier (Categories, elva blåa och fyra gula) samt markerade med en flagga utifrån den tidstypen (grön, blå eller gul) de tillhör. Denna klassificering har jag använt vid uppdelning på de olika flikarna.



Figur 18 - Exempel på tidskategorierna

8.2.2 Drill-down på önskad storhet

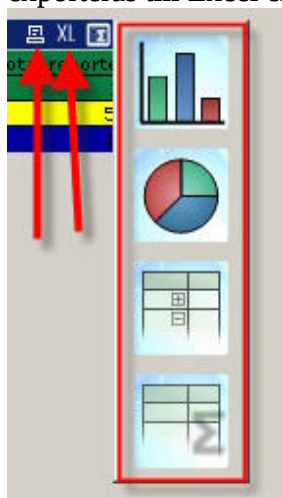
De storheter jag har använt som möjliga val i prototypen är : namn, user-ID, team (två olika sorter), tidstyp, projekt (3 nivåer) och kategori. Det är mycket enkelt att lägga till och ta bort önskade storheter.



Figur 19 - Exempel på hur valet av storhet ser ut i QlikView-rapporten

8.2.3 Ett antal olika visningsmöjligheter

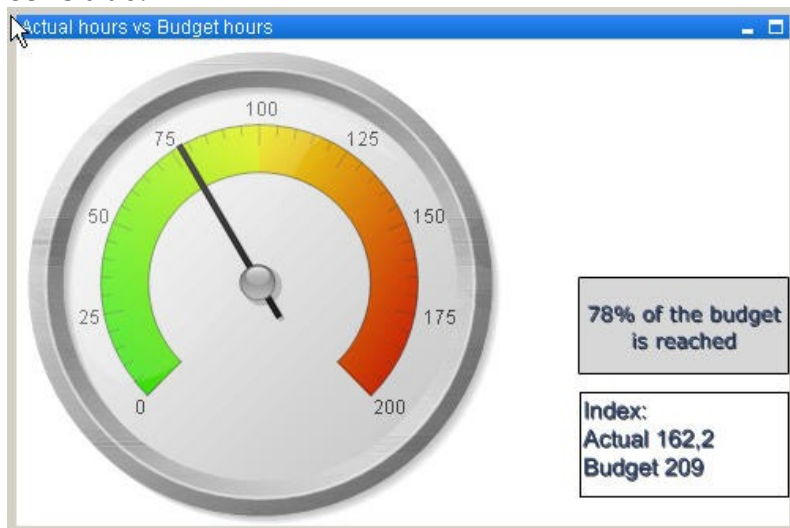
Det finns möjlighet att se informationen på olika sätt. Jag har valt cirkel- och stapeldiagram samt tabellform som default, samt gjort snabbkommandon för alla tabeller och diagram att exporteras till Excel eller skrivs ut.



Figur 20 - Snabbkommandon för print och Excel-export samt val av tabell- eller diagramformat

8.2.4 Budgetjämförelse

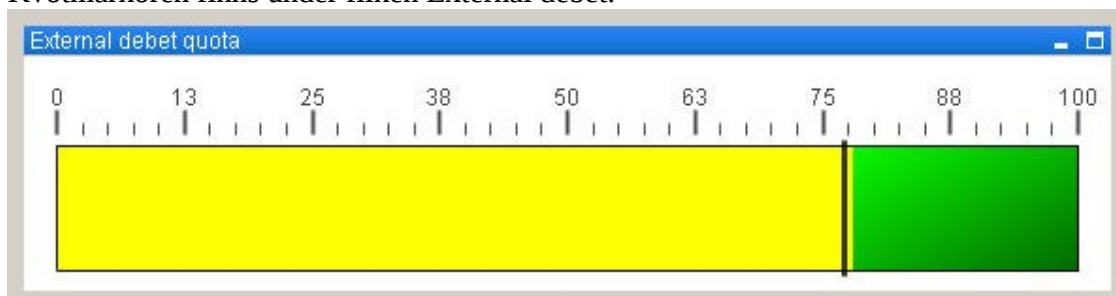
På budgetfliken finns en mätare som anger hur många procent av de budgeterade timmarna som uppnåtts. Det anges även exakt antal budgeterade timmar samt registrerat antal timmar. Siffror finns för de kategorier som funnits budgeterade i detalj, vilket har funnits för gul och blå tid.



Figur 21 - Exempel på budgetmätare för den blå kategorin IGEM där 78% av budgeten uppnåtts

8.2.5 Extern debiteringskvot

Den korrekta beräkningen av extern debiteringskvot görs genom att ta andel rapporterad tid ut mot kund (grön Follow Up-tid) av total rapporterad tid (IFS-tid). IFS tiden har inte varit tillförlitlig i applikationen så jag har räknat ut extern debiteringskvot som andel rapporterad tid ut mot kund (grön Follow Up-tid) av total rapporterad Follow Up-tid. Detta har jag illustrerat i en mätare. Företagets budgeterade kvot är markerad och delen under kvoten är gul och delen över budgeterad kvot är grön (då läggs mer tid än budgeterat mot kunderna, den vinstdrivande verksamheten). Aktuell kvot markeras med ett svart streck. Kvotmarkören finns under fliken External debet.



Figur 22 - Exempel på ett team som ligger något under budgeterad debiteringskvot

9 Diskussion

Arbetet med kravanalys och prototypbygge har varit intressant och lärorikt. I diskussionen tar jag upp styrkor och svagheter med prototypen och två punkter som varit problematiska under arbetets gång, och som jag anser kommer att påverka en framtida lösning. Jag diskuterar även fördelar och nackdelar med att genomföra lösningen skarpt samt för en kortfattad diskussion kring ekonomiska aspekter. Jag vill understryka att de åsikter som framförs i diskussionen är mina egna och har framkommit som en bearbetning av det arbete jag bedrivit.

9.1 Prototypens styrkor

En prototyp är inte tänkt att fungera skarpt. Jag tycker att den aktuella prototypen löser uppgiften att ge en bra bild av hur ett verktyg i QlikView kan komma att se ut, vilket var syftet med att bygga den. Detta ser jag som en mycket stor styrka.

Utöver detta så kan prototypen fungera som inspiration och en handfast startpunkt för vidareutveckling mot en färdig applikation. Den kan också användas för att förankra idén hos berörda parter och fungera som beslutsunderlag. En annan praktisk detalj är att man, vid eventuell vidareutveckling, kan genomföra användartester på prototypen för att få djupare förståelse för användarnas behov

9.2 Prototypens brister

Det finns fel i summeringen av vissa tidstyper jämfört med nuvarande rapporter. IFS-tid och Scorecard-tiden är felaktiga, med en felmarginal på runt 10%⁶. Jag tror att dessa tidsskillnader går att komma åt om man inför några av de avgränsningar i form av undantag som jag valt att inte ta in i prototypen. När dessa tidstyper stämmer bättre kan man även inför en slack-variabel, alltså en variabel för skillnaden mellan IFS-tid och rapporterad tid (alltså mellan faktiskt tid på jobbet och rapporterad tid).

OH-resurserna är i nuvarande lösning en egen grupp som kan rapportera tid på antingen grön, gul eller blå tid. Enligt en specifiering är det endast intressant att se den tid som OH-resurser lägger på extern tid, alltså den gröna tiden.

Många grupperingar och definitioner har gjorts ”i koden”, vilket gör att man måste ändra koden för att ändra exempelvis tidstyperna eller andra detaljer. I en framtida lösning bör så mycket som möjligt av dessa klassificeringar flyttas ut till externa dokument. Detta har idag endast gjorts för budgetsiffror samt alla OH-resurser.

9.3 Tidsrapporteringsystemet som allt bygger på

Tidsrapporteringsystemet Projus som är datakällan till nästan all data i systemet är byggt in house av SSD. Systemet har, sedan det byggdes, byggts ut ett antal gånger efter att man önskat ytterligare funktionalitet. Men för att kunna användas i den applikationen som jag tittat på och testat finns det viss funktionalitet som inte är så flexibel, och vissa

⁶ IFS-tid enligt Excel = 30756, enligt QV-prototyp = 28632. Diff = 6,9 %
Scorecard-tid enligt Bex = 29869, enligt QV-prototyp = 33598. Diff = 11,1 %

variabeltyper som inte är möjliga, och därför har ett antal problem behövt lösas. I förlängningen bör källan av tidsrapporteringsstrukturen ses över i avseende att bli så kompatibel mot den faktiska verksamheten som möjligt. De kritiska framgångsfaktorer som gäller QlikView-applikationen kan också appliceras på Projus, och att systemet ska vara verksamhetsdrivet är en central punkt.

9.4 En komplicerad verklighet

Verkligheten är de facto inte ett välordnat system av ekvationer som kan kodas in i ett digitalt system av ettor och nollor. Gällande tidsrapportering och arbetstider så sker det ständigt förändringar. Exempel kan vara att folk blir befordrade, byter avdelningar, plötsligt slutar, eller byter bolag. Även arbetsuppgifter kan skilja sig från "hur det är programmerat att vara", projekt kan till exempel vara interna trots att de har en extern projektkod eller vara ett projekt som ligger kvar sedan gammalt som ett utbildningsprojekt, men har nu bytt karaktär. Ett system kan bara vara en modell av verkligheten, där modellen försöker att efterlikna originalet i möjligaste mån. Undantag kan hanteras i enstaka fall, och därigenom ge en så komplett bild av verkligheten som möjligt. Men som sagt så kommer inget datorprogram att vara en exakt kopia av verkligheten, och så inte heller den aktuella prototyp eller en framtida vidareutvecklad applikation.

Verkligheten är alltså till sin natur komplicerad och inte kan beskrivas perfekt av datorer. Tidssystemet som tar in grunddaten är oflexibelt och kan inte, på ett optimalt sätt för just denna QlikView-lösning, hantera informationen. En applikation som bygger på en så komplicerad verklighet och oflexibel grunddata aldrig kan bli helt valid.

9.5 Fördelar med en framtida QlikView-lösning

En QlikView-applikation löser de flesta av de problem man hade med dagens rapportform (beskrivet i kapitlet om *Förutsättningar för en ny lösning*). Man har möjlighet att ha all information som fanns i den tidigare rapporten, men med väldigt utökade möjligheter.

Det är **lätt att växla mellan 'högt och lågt'**, alltså de stora perspektiven och de små detaljerna. Över lag finns stor flexibilitet i användandet.

Det finns **fler perspektiv** att titta på. Man kan välja flera olika utgångsvariabler såsom person, team, projekt och tidstyp. Detta för att applikationen kan ha ett bredare användningsområde och kan vara intressant för exempelvis teamchefer eller projektledare som vill följa upp sin verksamhet.

Det är **lätt att hitta enskilda poster**. Detta gör att programmet skulle kunna vara effektivt för problemlösning, och att reda ut olika tveksamheter i personalarbetet, en funktion som inte alls finns idag.

Tanken är att de kontinuerliga dataladdningarna kommer att **triggas automatiskt** vilket är helt skilt från hur det går till idag. Detta gäller dock inte undantag och specifika ändringar som måste skötas av en systemförvaltare. En smakfråga är att det är **roligare att använda** programmet.

9.6 Nackdelar med en framtida QlikView-lösning

Systemet **kräver en viss kunskap** om datat för användning. Man bör känna till namnbruket inom företaget (såsom team, projekt och resurser) och även den grundläggande strukturen kring tidskategorisystemet. Den kunskapen har de tilltänkta användarna idag (ledningsgruppen), men vid en ökad användarkrets bör man tänka på frågan.

Lösningen kommer **fortfarande att vara beroende** av BW-laddningar från Projus för att kunna uppdatera, vilket sker ett specifikt datum, efter att Projus har låsts för ändringar.

Lösningen kommer även fortsättningsvis att kräva viss handpåläggning, det går inte att automatisera hela lösningen. Detta gäller förändringar som uppkommer löpande.

9.7 Ekonomiska aspekter

Dagens kostnad för rapporteringen består i den arbetstid som krävs för Excelarbetet. DAF beräknar tidsåtgången till en dryg arbetsdag i månaden, och till det kan läggas några timmar som går åt till den andra rapporten, den som går ut till teamcheferna. Utöver detta används ca en timme åt att jämföra dessa två rapporters siffror.

Kalkylerad kostnad för en ny lösning är dels tid för nyutveckling och lansering av applikationen. Utöver detta kommer ett visst systemunderhåll att krävas, vilket är svårt att uppskatta precis, det beror på hur mycket prototypen kan förbättras. Jag har gjort en uppskattning att det skulle kunna röra sig om 2h per laddning, per månad.

Excelrapporter: (drygt) 1 dag/månad [$\sim X$] om X motsvarar antal dagar/månad

QV-rapporter: Utveckling Y dagar, samt underhåll 2 timmar/månad [$\sim Y + 0,25 \cdot X$]

Break even nås vid $X = Y + 0,25X$, vilket innebär att det, beroende på vidareutvecklingstid, kan ta ett tag innan alternativinvesteringen blir lönsam. Men i beräkningen ska också tas hänsyn till att man i och med dessa kostnader inte bara får de rapportsiffror som tidigare finns, utan även utökade användningsområden och flera potentiella användare. Detta är något jag tror att Sandvik Systems Development kan få stor nytta av och något jag tror är viktigt att beakta.

10 Sammanfattande rekommendationer

Slutsatserna i arbetet går att läsa kontinuerligt egentligen från nulägesanalysen och framåt, eftersom arbetsgången i sig innefattat ett antal slutsatser för vidare arbete. Under denna rubrik avser jag att kortfattat sammanfatta mina slutsatser för vidare arbete. Detta för att visa på hur arbetet är praktiskt tillämpbart.

10.1 Rekommendation till Sandvik Systems Developments ledningsgrupp

Under mitt arbete har jag funnit ett antal fördelar och några nackdelar med att förändra dagens rapportform. För att ta ett beslut om vidare arbete bör ni främst fundera över vad ni är intresserade av. Dagens rapporter ger en översikt över den information ni är primärt intresserade av. Det är bara ledningsgruppen som använder den rapport som DAF sammanställer.

Om ni inför en QV-lösning kommer ni att kunna utöka användarna av rapporterna, och detta måste ses över vid vidareutveckling. Att få fler användare och ökade möjligheter att få ut information man är intresserad av kring verksamheten är, enligt min åsikt, den stora vinsten av att införa applikationen skarpt. Jag är övertygad om att det går att få all funktionalitet som finns i dagens rapporter, samt även ett stort antal nya möjligheter.

10.2 Följder vid ett eventuellt införande

Om ni väljer att vidareutveckla mot en färdig QV-applikation vill jag rekommendera er att

- Utse tydliga ansvarsområden, både utvecklings- och förvaltaransvar
- Formulera hur applikationen bäst stödjer er vision, era mål och era processer
- Ställ krav på den funktionalitet ni behöver
- Förankra beslutet om applikationen hos berörda parter
- Identifiera vem utöver ledningen som ska använda applikationen, min rekommendation är att fler än idag bör dra nytta av den. Exempelvis team managers, service managers samt eventuellt project managers
- Avsätt tid för utbildning bland de identifierade användarna
- Ni jämför ihop budget och tidskategorier efter de behov ni har att följa upp tidskategorier jämfört med budget. Detta bör göras på ett mer realistiskt sätt än idag⁷.
- Använda applikationen då den är färdig

⁷ Totalbudget delas på 11 och juli nollbudgeteras

11 Källförteckning

11.1 Publicerat material

Anthony, R, Govindarjan, V (2007). Management Control Systems, 12th ed., McGraw-Hill, New York

Elmasri, R, Navathe, S (2004). Fundamentals of Database Systems, 4th ed., Pearson/Addison-Wesley Education, Boston

Holopainen, S, Lillrank, P, Paavola T (2001). Linking IT to Business – a Tale of Discovering IT Benefits, Studentlitteratur, Lund

Inmon, W.H (1992). Building the Data Warehouse, QED Technical Publ. Group, Boston

Kahaner, L (1996). Competitive Intelligence: how to gather, analyze, and use information to move your business to the top, Simon & Schuster, New York

March, J (1999). The Pursuit of Organizational Intelligence, TJ International, Padstow

McCarthy, T, Risch, T (2005). Databasteknik, Studentlitteratur, Lund

Sandström, B (2004). Att lyckas som operativ ledare, Industrilitteratur, Linköping

Saxby, S (1990). The Age of Information: The past development and future signification of computing and communications, Macmillan, London

Söderström, P (1997). Datalager Verksamhet, Metod, Teknik, Studentlitteratur, Lund

11. 2 Opublicerat material

Forsberg, M, Morell, M, (2006). Optimering av SQL-frågor för analys i QlikView, examensarbete på C-nivå, Karlstadsuniversitet

Strömberg, D (2006). Analys av BI-system och utveckling av BI-applikationer, examensarbete på D-nivå, Karlstads Universitet

11.3 Elektroniska källor

Sandviks hemsida. Fakta i korthet, www.sandvik.se (2007-08-13)

Sandviks hemsida. The Sandvik IT Portal (2007-08-13)

Sandvik Systems Developments intranät (2007-08-13)

QlikTechs hemsida. www.qliktech.se (2007-09-26)

11.4 Muntliga källor

Gun Andersson, 2007-05-10

Björn Lindén, special services manager 2007-05-15, 2007-05-28, 2007-08-10

Eva K Larsson, ekonomichef 2007-05-24
 Erik Pihl, business controller 2007-05-31 samt 2007-09-14
 Anna-Lena Söderberg, application manager 2007-06-04
 Kjell Persson, team manager 2007-06-07
 Erik Ljungberg, team manager 2006-06-11
 Lena Josefsson, QlikView-expert löpande juli-september 2007
 Linda Fernqvist, systemutvecklare, löpande under perioden maj-oktober 2007
 Johan Palm, team manager 2007-09-10

11.5 Figurförteckning

Figur 1- IT-koordineringen på Sandvik	8
Figur 2 – Underrättelsecykeln	10
Figur 3 - Schematisk skiss av en stjärnstruktur	13
Figur 4 - Schematisk skiss av snöflingestruktur	14
Figur 5 - Exempel på BEx-rapport <i>Project Follow-Up</i>	24
Figur 6 - Exempel på flikrad i Excel-rapport	24
Figur 7 - Intern tid och sammanställning av gul tid i Excel-rapporten	25
Figur 8 - Sammanställning i Excel-rapport	26
Figur 9 - Diagram i Excel-rapport	27
Figur 10 – Schematisk skiss över dataladdningarna	28
Figur 11 – Stjärnstruktur för Follow Up-tid	30
Figur 12 - Tabellkopplingar i QlikView.	35
Figur 13 - Flikraden i QlikView	35
Figur 14 - Exempel på hur tidstyper filtreras i tabelldefinitionen	36
Figur 15 - Ett utdrag av instruktionsfliken i QlikView	36
Figur 16 - Närbild på vänstermenyn i QlikView-rapporten	37
Figur 17 – Valmöjligheter i QlikView-rapporten	37
Figur 18 - Exempel på tidskategorierna	38
Figur 19 - Exempel på hur valet av storhet ser ut i QlikView-rapporten	39
Figur 20 - Snabbkommandon för print och Excel-export samt val av tabell- eller diagramformat	39
Figur 21 - Exempel på budgetmätare för den blå kategorin IGEM där 78% av budgeten uppnåtts	40
Figur 22 - Exempel på ett team som ligger något under budgeterad debiteringskvot	40
Figur 23 - Illustration av dataflödet vid laddning	48

11.3.1 Källförteckning för figurer

För de figurer där ingen källa anges är källan den rapporten som figuren är tagen från, alltså DAF-rapport för Excel eller egen för QlikView.

Figur 1. Sandviks intranät, The Sandvik IT Portal, 2007-08-13
 Figur 3. Egen bearbetning av Sundström, s 135
 Figur 3. Egen bearbetning av Söderström, s. 63
 Figur 4. Egen bearbetning av Söderström, s. 64
 Figur 21. Egen bearbetning av Strömberg 2006, s16

12 Bilagor

Förteckning av bilagor

I de fall då källan inte är en produkt av examensarbetet anges källa inom parantes, där efternamn refererar till de respondenter som återfinns under muntliga källor.

1. Teknisk information om QlikView
2. Kravspecifikation
3. De olika tidskategorierna (Ljungberg) – **Borttagen ut denna version**
4. Tabeller och fält som laddas till QV från BW (Fernqvist) – **Borttagen ut denna version**
5. Detaljerad stjärnstruktur i BW (Fernqvist) – **Borttagen ut denna version**
6. Scriptkod (egen, med hjälp av Josefsson) – **Borttagen ut denna version**
7. Intervjumall

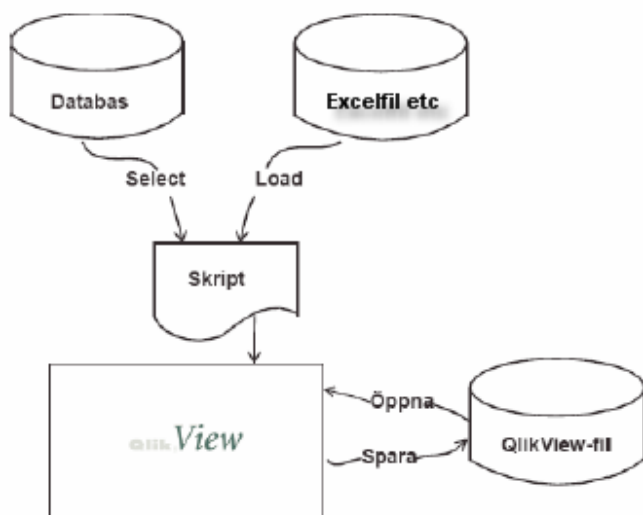
Bilaga 1- Teknisk information om QlikView

Datamängden i det befintliga datalagret (BW) är sparad i en kubstruktur. Tyvärr är kubstrukturen svår att hämta data ur på ett strukturerat sätt, rent teknisk. Därför omformas strukturen, och man lägger in datan i en s.k. ODS (operational data store), ett lagringsformat där alla fält har ett tekniskt namn (ex [/BIC/ZIS00048W]). Från ODS:en kan man sedan via en SAP-connector läsa in strukturen till QV. Connectorn behöver en så kallad FileClient för att kunna läsa över datat.

De data som önskas specificeras i SQL-filer, som består av enkla SQL-queries (av typen SELECT 'ODS-fältnamn' FROM 'ODS-tabellnamn'). FileClient läser från SQL-filerna och skapar en qvd-fil som motsvarar varje sql-query. Dessa laddar man sedan in i själva QlikView-programmet, i ett första steg.

I QlikView bygger man applikationer genom att man läser in datan som ska användas via ett script. Scriptet kan man läsa in med egenskriven kod, eller så kan man ladda in datan via färdiga strukturer, till exempel qvd- eller xls-filer. I det första steget laddar man alltså in de qvd-filer som skapats i ett QV-script. I nästa steg (själva applikationen) måste man "ordna upp" i datastrukturen. QV fungerar så att alla tabeller är kopplade, med en (och endast en) nyckel. Om det finns multipla nycklar så kopplas dessa ihop till en sammansatt nyckel. Det får inte heller finnas mer än en variabel med samma namn, på de ställen där det finns samma namn (date, hours, amount och liknande) måste dessa döpas om eller modelleras om för att accepteras. Alla sådana här ändringar görs i det aktuella scriptet, och sparas sedan, och laddas om.

När scriptet är färdigt, och alla data är laddade kan man börja modellera med datamängderna. Beroende på vilken information man är intresserad av kan man göra olika typer av beräkningar eller grafiska element. Det kan till exempel röra sig om statistiska beräkningar, olika grafer eller summeringar.



Figur 23 - Illustration av dataflödet vid laddning

Bilaga 2 - Kravspecifikation för en Qlikview-applikation

Funktionell beskrivning

QlikView har olika worksheets, liknande Excel. Resultatet ska bli ett antal funktionella worksheets. Flikarna är tänkta att representera:

- En flik för grön tid: extern debitering och OH-debitering
- En flik för blå tid: interna projekt och uppgifter
- En flik för gul tid: intern tid övrigt, uppdelat i ett antal kategorier
- En flik för sammanställning av de tre, här ska man få en överblick över hur verksamheten går
- En flik för budgetuppföljning för de siffror som finns tillgängliga
- En flik för instruktioner

Alla flikar har samma utsökningsgrunder, nämligen:

- Tid, år och månad
- Team (ca 15 st)
- Projects (runt 100 st)
- Resources (>300)

Hela SSD samt hela innevarande år är default-sökningen, alltså startvärde. (I prototypen finns dock enbart data från maj 2007)

Konsekvenser

Tillgänglighet

Applikationen läggs på den interna Sandvik-QV-portalen. Där kommer de tillåtna användarna åt QV antingen som webbdokument (rekommenderat) eller via att de har QV installerat (ej rekommenderat)

Påverkan på befintligt rapportsystem

Eftersom vi laddar över data från BW direkt till QV kommer inte det nuvarande Excelformatet att påverkas alls. Man kan fortsätta med den befintliga rapporteringen som förut. Om försöket med QV skulle falla väl ut finns det god potential att utveckla QV-applikationen skarpt. Då måste alla de undantag som hanteras i dagens rapporter men inte i prototypen föras in på ett smidigt sätt i den nya applikationen.

Verksamhet

Tidsrapporteringen är en kritisk del av SSDs verksamhet. Eftersom rapporteringen utgör grunden för faktureringen till kund så är det grunden till hela den ekonomiska analysen. Att ha ett fungerande och lättanvänt tidsuppföljningsverktyg är därför helt avgörande för verksamheten. Verksamheten kommer dock inte påverkas nämnvärt av prototypen, annat än om man beslutar att vidareutveckla den skarpt.

Drift

Driftsproblemet är till största del okomplicerat, då uppdateringar laddas regelbundet från BW krävs ingen större insats. Det stora driftsproblemet består i de undantag som rapporten innehåller. Det finns ett antal avvikelser som ändras månadsvis i och med att verksamheten förändras. Dessa undantag måste matas in manuellt av någon driftsansvarig, som har just den kunskapen, samt kunskapen om hur man modifierar QV-applikationen. Idag görs dokumentet från grunden varje månad, och en person sitter och lägger in alla dessa undantag. Så jämfört med idag blir det en minskning av underhållsbehov, men det går inte att helautomatisera.

Utbildning

Ingen av de intervjuade användarna har någonsin jobbat med QV. De har dock mycket god datorvana. Detta innebär att man måste ha någon typ av inledande utbildning i hur man använder QV, och därefter borde de, om programmet är tillräckligt lättanvänt kunna navigera och använda det på egen hand.

Migrering av data

Hela applikationen bygger på att man kan överföra data mellan de olika systemen. Från BW tar man verklig produktionsdata och för över via ODS-format till QlikView. Där laddar man data i flera steg, i vilka man genomför olika typer av modellering där det sista steget är den slutgiltiga applikationen.

(Se bilaga med datafält för information om vilka data som ska laddas över från BW till QV)

Tolkning av användarkrav

Här beskriver jag de krav som användarna har, och hur jag tolkat detta i design och modellering, alltså i rent praktiska termer.

Ett lättanvänt format

De tidigare två rapportformerna var svåränvända av olika anledningar. Bex-rapporterna var dels svåra att få tillgång till, och sedan måste man lära sig hur man gör specifika utsökningar, vilket var för svårt om de bara används ca en gång per månad. På Excel-bladet var sökningarna färdiggjorda, och sammanställningarna var färglagda enligt den specifika färgkod som tagits fram. Excelbladet innehåller 17 flikar, av olika karaktär. Innehållet är svåröverblickbart.

En QV-lösning är användarvänlig på flera sätt. Man skapar själv designen, så det finns alla möjligheter att göra lösningen effektiv och lättanvänd. Man kommer även att ha tillgång till data i utvalda utsökningsrutor, vilket underlättar förståelse och användning.

Ett lättillgängligt format

QV-lösningen kommer att läggas som ett webbdokument på Sandviks interna QV-portal. Där uppdateras det löpande, antingen händelsetriggerat eller tidsstyrt. Man kommer då alltid åt de senaste siffrorna via ett enkelt web-gränssnitt.

Möjlighet till flexibilitet

Skillnaden mellan QV och Excel är enorm, gällande just flexibilitet. Eftersom QV är ett OLAP-verktyg så jobbar den mot det insparade datalagret. Detta ger möjlighet till mycket specifika drill-downs, då all data finns inläst, och finns tillgänglig genom de utsökningar man gör. Det är en design detalj vilka sökalternativ man väljer att ha, dessa kommer att tydliggöra strukturen.

Lätt jämförbart mot budget

Budgeten är bara att lägga in (från exempelvis befintliga Excel-dokument) som separata data, och markera i grafiken, så data kommer alltid att jämföras mot budgetvärden på de ställen där dessa finns.

Tydligt gränssnitt

Användarna har behov att kunna se både ”det stora och det lilla”, alltså se trender och sammanhang samtidigt som det finns intresse av att kunna drillar på enskilda ”problem poster”. Detta löser QlikView på ett bra sätt där användarens egna utsökningar bestämmer nivån på den visade informationen.

Det ska inte finnas olika bud om informationen

Med dagens spridda rapportformer, där det i nuläget finns minst fyra⁸ olika sätt att få fram liknande information där alla bygger mer eller mindre på varandra. Detta problemet överbygger man genom att ha en enda rapport. Siffrorna i rapporten kommer från en entydig datakälla, tabellerna i BW, och kan därför enkelt källgranskas om problem skulle uppstå.

⁸ Två excelrapporter, BEx-rapporter samt ”vanliga” utsökningar i Projus

Bilaga 7 - Intervjumall

Till vad (/hur) använder du rapporterna idag?

Vilken information är viktig för att du ska kunna göra bra bedömningar?

Vilken av den informationen finns samt finns inte på dagens rapporter.

Är det något annat du tänkt på som fattas på dagens rapporter?

Till vad skulle du vilja använda rapporterna?

Hur skulle rapporterna kunna utformas för att underlätta ditt arbete?

Vad på dagens rapporter är helt överflödigt?

Det finns ett förslag på en lösning där man kan göra utsökningar, vilket inte är möjligt idag.
Finns det något särskilt perspektiv du tycker skulle vara intressant för din verksamhet?
(projekt/leverans, enskilda personer etc.)

Hur skulle du vilja komma åt dem (intranät, epost??)

Datorvana, hur stor är din datavana inom:

	stor	viss vana	aldrig använt
SAP			
BW			
Excel			
Qlikview			