



Metodologianvändning och -anpassning i IT-branschen.

En empirisk studie av drivande faktorer för metodologianpassning i IT-företag.

Abstrakt

Uppsatsen behandlade organisationers anpassning av systemutvecklingsmetodologier till den dagliga verksamheten, samt vilka drivande faktorer som låg bakom dessa anpassningar. Empirin baserades på intervjuer med Accenture, ADB-kontoret, IFS, Intenia och WM-data, åtföljt av analys och extrahering av drivande faktorer. Uppsatsen inkluderar inte recension av metodologier, heller inte rekommendation av specifik metodologi. De intervjuade företagen delades in i följande kategorier: IT-konsultföretag med specifika allianser, IT-konsultföretag utan specifika allianser samt företag med egenutvecklad IT-produkt. Följande drivande faktorer identifierades: formaliserat partnerskap, förändrad affärsmodell, förändrade kundkrav, individuell kompetens, kostnadseffektivisering, organisationsförändringar, trender/konkurrenser, uppdragets variation samt verksamhetsinriktning.

Nyckelord: Anpassning, drivande faktorer, metod, metodologi, processer, systemutveckling

Författare: Björn Olsson , Mattias Grytting
Handledare: Maria Bergenstjerna
Magisteruppsats, 20 poäng

Innehållsföreteckning

<i>Executive summary</i>	4
<i>Inledning</i>	5
Bakgrund	5
Software crisis	6
Frågeställning	6
Avgränsning	6
Språk.....	6
Målgrupp	6
<i>Metod</i>	7
Tre olika forskningssynsätt	7
Positivism	7
Relativism.....	7
Socialkonstruktivism.....	7
Valt synsätt	7
Teoretisk datainsamling	8
Empirisk datainsamling.....	8
<i>Begrepp</i>	9
Anpassning	9
Metod	9
Metodologi	9
<i>Teori</i>	10
Systemlivscyklar	10
Sekventiell utveckling	10
Iterativ utveckling	11
Rapid prototyping.....	12
Synsätt på metodologianvändning	13
Prescriptive.....	13
Contingency	13
Method engineering.....	13
”Hybridmetoden” – Motorola	14
Methodologies/Procedures/ Techniques/Tools	15
Methodologies.....	15
Procedures	15
Techniques	16
Tools.....	16
Samverkan mellan Procedures, Techniques och Tools	16
Pyramiden.....	17
Perrows teknologitypologi	18
Hatchs femcirkelmodell	19
Egen förklaringsmodell	20
Process.....	20
Metodologi	20

Aktivitet.....	20
Interaktion med ”Pyramiden”.....	20
Egen modell för anpassningstypologier	22
Företag med egenutvecklad IT-produkt	22
IT-konsultföretag utan specifika allianser.....	22
IT-konsultföretag med specifika allianser.....	22
Spelföretag	22
<i>Empiri</i>	24
IFS	24
Beskrivning av metodologianvändande	24
Metodologianpassning hos IFS	26
ADB-kontoret.....	27
Beskrivning av metodologianvändande	27
Metodologianpassning hos ADB-kontoret.....	27
WM-data.....	28
Beskrivning av metodologianvändande	28
Metodologianpassning hos WM-data.....	29
Intentia.....	30
Beskrivning av metodologianvändande	30
Metodologianpassning hos Intentia.....	30
Accenture	32
Beskrivning av metodologianvändande	32
Metodologianpassning hos Accenture	33
<i>Analys och diskussion</i>	34
Indelning av företag	34
Analys av empiriskt material m.a.p användning och anpassning av metodologier	35
Verktyget blir metodologin	35
Kartläggning av drivande faktorer	36
Förklaring av drivande faktorer.....	37
<i>Slutsats</i>	42
Hur anpassar organisationer systemutvecklingsmetodologier?	42
Vilka faktorer driver anpassningen av systemutvecklingsmetodologier?.....	42
Reliabilitet och validitet	43
Reliabilitet	43
Validitet.....	43
Förslag till vidare studier.....	44
Spelföretag	44
Verktygen är metodologin.....	44
<i>Ordlista</i>	45
<i>Referenser</i>	48
<i>Bilagor</i>	51

Executive summary

Studien behandlar organisationers behov av användning och anpassning av systemutvecklingsmetodologier, samt vilka drivande faktorer som ligger bakom dessa anpassningar. Vi kan konstatera att olika organisationer ställer olika krav på systemutvecklingsmetodologier. En grundläggande indelning av dessa organisationstyper kan göras enligt följande:

- **Företag med en egenutvecklad IT-produkt**
Här återfinns företag som affärssystemstillverkarna IFS och Intenia. Dessa företag tenderar att skraddarsy sin metodologi till sin specifika produkt för att på detta sätt skapa en metodologi som fungerar mycket väl ihop med den egna produkten, men som inte längre är generisk.
- **Renodlade IT-konsultföretag**
I denna grupp ser vi företag som ADB-kontoret och WM-data. Dessa företag, vars uppdrag varierar kraftigt, har också ett stort behov av en metodologi som är så pass generisk att den går att anpassa på projektbasis efter givna förutsättningar.
- **IT-konsultföretag med strategiska allianser med mjukvarutillverkare**
Accenture är ett exempel på denna företagstyp, vars affärsmodell innefattar allianser med mjukvarutillverkare som t.ex SAP, Oracle och Microsoft. När verksamheten baseras på denna typ av allianser underlättar det om den för organisationen gemensamma systemutvecklings metodologin har specifikt stöd för implementation av varje partners produkt. Detta får till följd att metodologin finns i flera instanser, en skraddarsydd instans för varje typ av implementation, men även i ett generiskt standardutförande.

De faktorer som författarna fann vara drivande i anpassningen av systemutvecklingsmetodologier är följande, presenterade i alfabetisk ordning:

- **Formaliserat partnerskap**
- **Förändrad affärsmodell**
- **Förändrade kundkrav**
- **Individuell kompetens**
- **Kostnadseffektivisering**
- **Organisationsförändringar**
- **Teknologiförändringar**
- **Trender/konkurrenser**
- **Uppdragets variation**
- **Verksamhetsinriktning**

För en detaljerad beskrivning av ovanstående faktorer hänvisas till resultatavsnittet. Slutligen vill vi bara påpeka att en metodologi är normativ och skall hanteras därefter, inte heller är en metodologi ett självändamål. Något som ibland kan få erfara bittert är att användarna blir slavar under metodologin, syftet blir inte att tillverka ett fulländat system utan att följa metodologin. Dessa två påpekanden illustreras väl av följande citat:

”...jag tycker att en metod är normativ det är otroligt viktigt... den beskriver normalfallet...”
IFS

”En metodik lever aldrig enbart, bara metodiken gör inga underverk.”
WM-data

Inledning

Vårt personliga intresse för anpassning av systemutvecklingsmetodologier bottnar i det faktum att vi under vår studietid har använt ett antal metoder/metodologier i kursverksamheten. Dessa metodologier har använts för att ta fram mindre, hypotetiska system för en på papper fördefinierad verksamhet. Vi inser och anser att en systemutvecklingsprocess skall stödja den befintliga verksamheten om denna verksamhet inte är definierad på papper utan är en faktisk verksamhet, inbegripande människor, datorer och informationsprocesser. Krav kommer följdaktligen att ställas på metodologins anpassningsbarhet till denna verklighet. Med ovanstående förutsättningar faller det sig naturligt att ställa frågorna: Hur anpassas systemutvecklingsmetodologierna och vad är det som driver denna anpassning? Akademiens intresse för detta ämne bottnar i behovet av att få ta del av hur systemutvecklingsmetodologier används och anpassas i verkligheten. Frågeställningen är även intressant för näringslivet för att kunna skapa en så god metodologi som möjligt som är anpassad efter verksamheten.

Bakgrund

Under de dryga 40 år som systemutveckling har funnits som disciplin har systemutvecklingen genomgått ett antal utvecklingsfaser, vilka kort kan sammanfattas på följande sätt:

1960- och 1970-talet definierades av avsaknaden av metodologier och fokus låg istället på att lösa framförallt hårdvarubaserade tekniska problem (Avison D. E, 2003).

1970- och 1980-talet kännetecknas av ett framväxande metodtänkande som resulterade i bland annat livscykelmodellen (SDLC) och indelning av utvecklingsarbetet i faser (Avison D. E, 2003).

Sent 1980-tal till sent 1990-tal karaktäriseras av ett utbrett metदानvändande och en ständigt ökande metodologiflora.

I samband med den ökade formaliseringen samt större synligheten av systemutveckling uppmärksammades också problem vad gäller kostnad, kvalitet och tid.

Generellt har dessa problem sammanfattats i uttrycket "the software crisis" (se nedan); d.v.s systemen kostar för mycket, tar för lång tid att utveckla och håller undermålig kvalitet m.a.p insatta resurser (Fitzgerald B, 1996). En del organisationer som sysslar med systemutveckling har delvis på grund av denna diskussion, för att undkomma problemen, omvärderat sitt synsätt på systemutvecklingsmetodologier.

En konsekvens av denna omvärdering är att en del organisationer har slutat att använda s.k "out of the box"-metodologier, d.v.s heltäckande metodologier där upphovsmännen/kvinnorna hävdar att metodologierna bör följas till punkt och pricka, i en given ordning för att ge önskade resultat (Avison D. E, 2003). Behovet av utvecklingsmetodologier för att strukturera verksamheten kvarstår, därför har intresset ökat för nyutvecklade tankesätt såsom "method engineering" (Funk P. J, 2000), "agile methods" (Highsmith J, 2002) och "dsdm" (DSDM, 2004).

Kännetecknande för dessa tankesätt är en öppenhet för användning av flera olika delar av olika metoder/metodologier för att skraddarsy en ny metodologi för organisationen som helhet eller för ett enskilt systemutvecklingsprojekt.

Software crisis

Denna term som här ovan sades beteckna en problematik inom systemutveckling där kostnad, tid och kvalitet är faktorer vars mål sällan eller aldrig uppnås är omdebatterad. Enligt studier uppnådde enbart 28% av IT-projekten de i förväg uppsatta kraven gällande kostnad, tid och kvalitet. 23% av projekten implementerades aldrig eller avslutades innan utvecklingen var färdig. 49% av projekten färdigställdes, dock med högre kostnad än budgeterat, efter utsatt tid och utan samtlig specificerad funktionalitet (Holtem R, 2001). Detta kan kontrasteras med DeGrace och Stahls uppfattning att problematiken är mycket överdriven och egentligen bara är en anledning till att försöka hitta nya systemutvecklingsstrategier (DeGrace & Stahl, 1990).

Frågeställning

Hur anpassar organisationer systemutvecklingsmetodologier?
Vilka faktorer driver anpassningen av systemutvecklingsmetodologier?

Det är viktigt att påpeka att vi även kommer att se val och byte av metodologi som en typ av anpassning då detta i sig självt är en anpassning till organisationens förutsättningar.

Avgränsning

Vi har för avsikt att granska anpassning av systemutvecklingsmetodologier, vilket innebär att vi inte kommer ha som mål att granska metodologier som har som huvudmål att göra organisationsförändringar eller strategiska affärsförändringar, t.ex Business Process Reengineering, BPR, eller metodologier som främst inriktar sig på styrning av projektprocesser.

Det är inte heller vår mening att rekommendera någon specifik metod/metodologi/process.

Språk

Då större delen av den litteratur som vi inhämtar vår kunskap ifrån är skriven på engelska har vi valt att, i den mån vi inte hittar någon vedertagen, klart likalydande svensk term, behålla originalspråkets ordval. Detta för att inte förlora i kommunikativ precision.

Målgrupp

Vår avsedda målgrupp, till vilka denna uppsats riktar sig, är individer som har god insyn i den systemvetenskapliga kontexten. Vi har därför inte för avsikt att på ett djuplodande sätt utveckla varje för kontexten hemmavarande term utan kommer istället att arbeta med referenser. Vi uppmanar därför de läsare som känner ett behov av djupare kunskap att söka densamma i de referenser som lämnas.

Metod

Detta avsnitt inleds med en beskrivning av tre vanliga forskningssynsätt. Under rubriken "valt synsätt" presenterar författarna sin syn på kunskapsbildning mot bakgrund av tidigare presenterade forskningssynsätt. Därefter presenteras tillvägagångssättet för insamling av material, både teoretisk litteratur och empiriskt material.

Tre olika forskningssynsätt

Positivism

Positivismen utgår från ett antagande om att det finns en universiell ofärgad sanning. För en positivistisk forskare gäller det att hitta denna sanning. Detta görs framförallt genom kausala samband och hypotesbyggande. För att fastställa samband och hypoteser används till största delen mätningar och experiment. För att etablera ett kausalt samband måste resultaten från mätningar och experiment vara upprepbara (Easterby-Smith, Thorpe & Lowe, 2002, s.27 ff). En positivist ser det som möjligt för forskaren att stå fri från det han eller hon studerar, d.v.s att forskaren i sig och själva studerandet inte påverkar objektet som studeras.

Relativism

Det relativistiska synsättet bygger på en uppfattning om att det inte finns någon universell mätbar sanning, utan de "fakta" man rör sig med alltid är tolkade av någon/några (Easterby-Smith, Thorpe & Lowe, 2002, s.27 ff). Dessa individer har givetvis en egen världsbild som har färgat deras syn på dessa fakta.

I tillägg utgår det relativistiska synsättet från ett ontologiskt antagande om att en viss situation existerar med vissa förutsättningar. Detta antagande utgör grunden i forskningsfrågeställningen och skall med hjälp av olika tekniker vidimeras och utredas. Denna vidimering och utredning bedrivs ofta med hjälp av mönsterpassning och korrelation, med bland annat intervjuer och enkäter som redskap.

Resultatet av ovanstående process blir ofta en ökad förståelse av forskningsproblemet som i sin tur bidrar till att öka forskningsdisciplinens kunskapsmassa.

Socialkonstruktivism

Detta synsätt bottnar i en önskan att förstå situationer och sociala samband. Till skillnad från t.ex. positivismens mätningar och experiment används här kommunikation som det främsta medlet för att nå målet. Socialkonstruktivismen ger inte forskaren någon möjlighet att distansera sig eller stå fri från forskningssituationen. Socialkonstruktivismen anser också att kommunikationen är det medel med vilket både forskare och påforskade upprättar de meningssamband som utgör byggstenarna för förståelse (Easterby-Smith, Thorpe & Lowe, 2002, s.27 ff).

Valt synsätt

Vår ingångsvinkel m.a.p kunskapsbildning är relativistisk med en dragning åt socialkonstruktivism (se ovan) d.v.s vi erkänner tolkningens betydelse och närvaro. Allt är situerat och färgas av både forskare och påforskade. Detta innebär att eventuella resultat med nödvändighet har en begränsad giltighet i forskningsdiskursen, och bör användas därefter. Dock, med detta i åtanke har undersökningen utformats på ett sätt som enligt vår mening medger någon grad av generaliseringsmöjlighet, då intervjuerna är spridda över ett antal individer inom någorlunda likartade verksamhetsområden.

Teoretisk datainsamling

Kunskapsinhämtning från befintlig kunskapsmassa utfördes i huvudsak med hjälp av för forskningsdiskursen relevanta artiklar som varit publicerade i erkända journaler. Ekonomiska biblioteket vid Göteborgs Universitet (Ekonomiska biblioteket, 2004) har varit den främsta källan i detta arbete. Ekonomiska biblioteket ledde oss till den elektroniska artikeldatabasen ACM (ACM, 2004). Vi har även haft förmånen att få ta del av kunskap som finns inom ramen för Informatikinstitutionen vid Göteborgs universitet samt utomstående välrenommerade forskare vid Georgia State University.

Empirisk datainsamling

I enlighet med vårt relativistiska forskningssynsätt anser vi att semistrukturerade intervjuer, d.v.s intervjuer med öppna diskussionsfrågor, är ett gott kvalitativt verktyg för öka kunskapen inom problemområdet, då denna typ av intervjuer har tillräckligt hög frihetsgrad för den intervjuade att styra en diskussion dit hon eller han finner lämpligt. På detta sätt anser vi oss kunna extrahera mesta möjliga material ur intervjun, oavsett vilken mental distans den intervjuade har till den av organisationen använda systemutvecklingsmetodologin.

Intervjuerna har genomförts med personer från fem olika företag, de intervjuade har god insyn i systemutvecklingsprocessen på respektive företag. Dessa intervjuer har tagit cirka två timmar i anspråk per företag och har bedrivits under perioden 17 Mars 2004 till 17 April. Intervjuerna har bandats och transkriberats i sin helhet. Intervjumaterialet kan vid första anblicken uppfattas som tämligen ostrukturerat, då det inte är formaliserat utan direkt transkriberat i sin givna form, d.v.s talspråk. Frågebatteri och intervjumaterial bifogas i bilaga 1 respektive 2 och framåt.

Begrepp

Detta kapitel behandlar i uppsatsen flitigt använda begrepp och deras definitioner.

Anpassning

Enligt Nationalencyklopedin (Nationalencyklopedin, 2004) betyder ordet anpassa *”att ordna (ngt) efter omständigheterna”*. Vi vill här förtydliga detta genom att förklara vad vi menar med ordet anpassning. Ovanstående förklaring är god men författarna vill tillägga att vi även ser byte och val av systemutvecklingsmetodologier som en typ av anpassning. Detta eftersom det faktiskt i enlighet med ovanstående är att ordna upp sin situation efter omständigheterna.

Metod

Nationalencyklopedins (Nationalencyklopedin, 2004) förklaring av ordet metod är följande: *”planmässigt tillvägagångssätt för att uppnå visst resultat”*. Detta stämmer gott överens med vår syn på vad en systemutvecklingsmetod/metodologi är, med tillägget som senare tas upp, att det går att tolka det planmässiga tillvägagångssättet på ett antal olika sätt, se avsnittet *”Synsätt på metodologier”* nedan.

Metodologi

Metodologi kan definieras och har definierats på ett otal sätt (Avison & Fitzgerald, 1995, s.418). Den springande punkten är ett för en metodologi specifikt filosofiskt perspektiv. Detta perspektiv torde utgöras av den *”ryggsäck”* av erfarenheter och syn på världen och dess funktion inneboende hos både utvecklaren av metodologin och hos användarna av densamma. Det finns, som vi uppfattat det, i huvudsak två sätt att se på detta filosofiska perspektiv. Det ena synsättet, som representeras av t.ex Avison och Fitzgerald (Avison & Fitzgerald, 1995, s.419), innebär att det filosofiska perspektivet s.a.s *”tillhör”* metoden och är oskiljbart från den. Det är detta som just gör metoden till en metodologi. Det andra synsättet, som delas av t.ex Checkland (Checkland P, 1981, s.162f), påpekar att perspektivet är en del av utvecklarens eller användarnas värld och således kan betraktas som skilt ifrån metoden, men i högsta grad påverkar utvecklingen och användningen av den.

Författarna anser att Avisons definition av metodologi är att föredra, med den skillnaden att det inte är möjligt, givet författarnas syn på forskningsmetodik, att skapa eller använda en metod utan att ha ett filosofiskt perspektiv. Detta perspektiv kan omöjligen undvikas, då det är en integrerad del av antingen upphovsmännen/kvinnorna eller användarna av metodologin. Detta ställningstagande innebär alltså att författarna anser att det inte överhuvudtaget existerar metoder, utan endast metodologier. Dock, för att vara så flexibla som möjligt i bl.a intervjudelarna av uppsatsen, används båda termerna. Då metod används som term skall den därför tolkas som metodologi.

Teori

Kapitlet kan grovt delas in i två delar. I första delen presenteras för uppsatsen relevanta redan existerande teorier. Kapitlet avslutas med författarnas egna förklaringsmodeller med bakgrund i det befintliga teoretiska ramverket.

Systemlivscykler

En systemlivscykel skall ses som den mest övergripande typen av formalisering. Eftersom det rör sig om en så övergripande typ av process är det vanligt förekommande att man inte specificerar exakt vad som skall utföras i varje namngiven fas utan snarare låter namnet på fasen tala för sig själv. Detta innebär naturligtvis att man lämnar fritt för den inom fasen tilltänkta metodologin att specificera detta. På detta sätt specificeras vad som skall göras men inte hur det skall utföras.

Sekventiell utveckling

Denna typ av livscykel består traditionellt sätt av ett antal faser där varje enskild fas' kriterier skall vara uppfyllda för att projektet skall få lov att ta steget till nästa fas. Ett klassiskt exempel på denna typ livscykel är vattenfallsmodellen (Avison & Fitzgerald, 1995, s. 17).

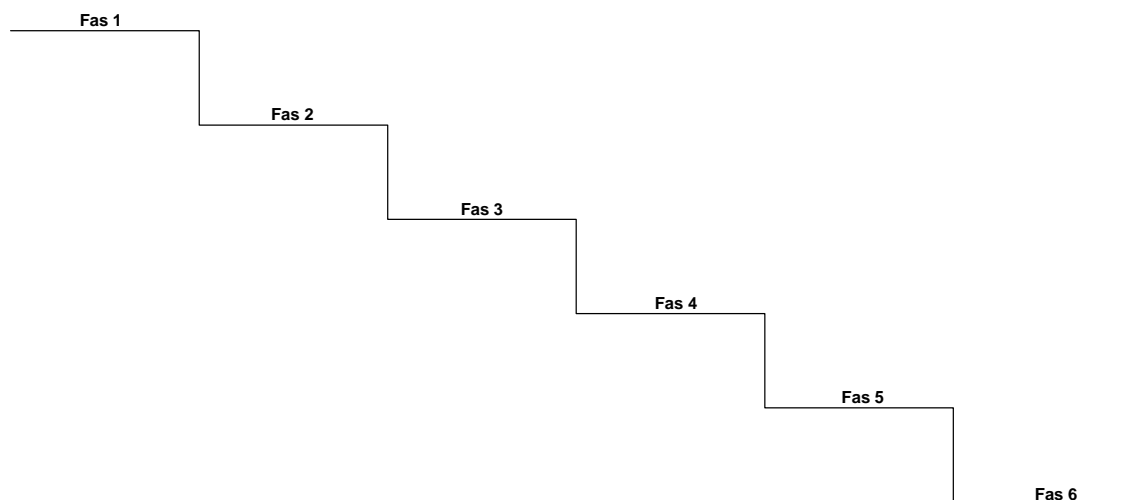


Fig. 1, Klassisk sekventiell utveckling

Modellen är en av de klassiska inom systemutveckling vad gäller sekventiell utveckling. Denna typ av utveckling innebär att fas ett måste vara färdigt innan fas två får lov att påbörjas. Ett annat karaktäristiskt drag består i att man inte itererar över faserna utan går från första fasen till sista utan återkoppling. Modellen har på grund av detta varit föremål för mycket kritik (Fitzgerald B, 1994). Kritiken baseras på den låsning som lätt uppstår då utvecklarna tenderar att i första fasen hämta in kundens krav och under resterande faser utveckla ett system, utan att under processens gång inhämta information om eventuellt ändrade krav och förutsättningar. Systemet valideras således inte mot kraven förrän i sista fasen, vilket innebär att möjligheten att under processens gång adaptera utvecklingen till en föränderlig verksamhet är liten.

Iterativ utveckling

Delar av den kritik som framförts mot den sekventiella livscykeln bygger på dess hårda faser (Fitzgerald B, 1994), d.v.s att när du har lämnat en fas så är den fasen låst och kommer inte att återbesökas. En så kallad lägesfrysning. För att komma ifrån detta förespråkas iterationer över faserna, att utvecklingen tar flera "rundor" över trappstegen. Dessa iterationer behöver då inte gå från första till sista trappsteget utan kan t.ex iterera över "Fas 3", "Fas 4" och "Fas 5" som figuren nedan illustrerar, för att vid uppfyllda krav gå vidare till "Fas 6".

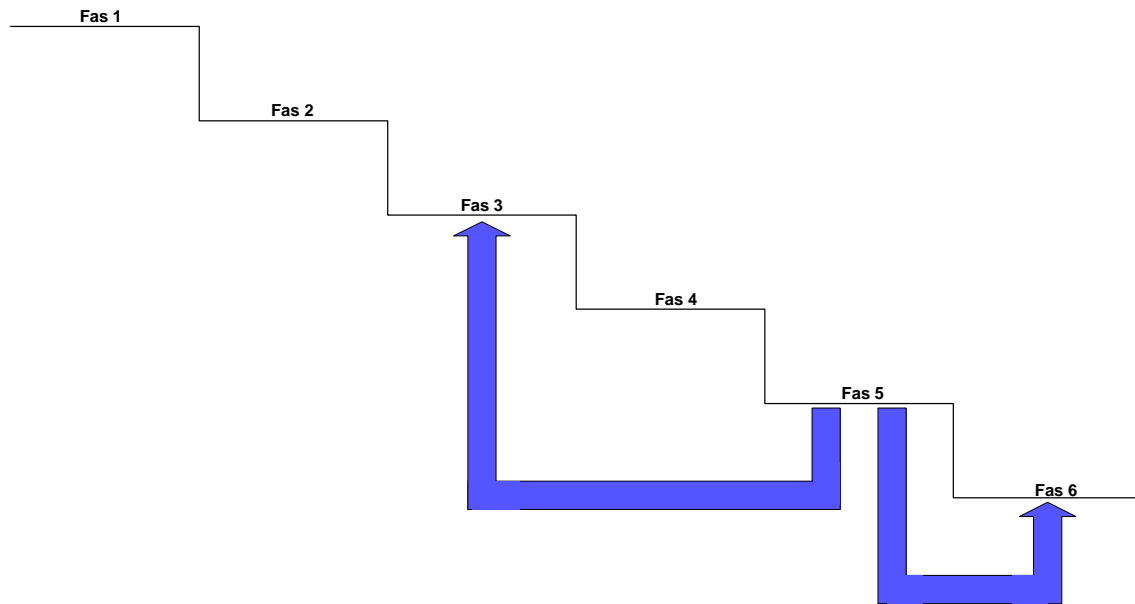


Fig. 2, Iterativ utveckling

Således finns möjligheten att gå tillbaka i utvecklingsfaserna. På detta sätt fångas eventuella nya och/eller förbisedda krav upp och steget ner till "Fas 6" tas i sedvanlig ordning när övriga faser är klara. Denna typ av återkoppling löser vissa av problemen med sekventiell utveckling men dock inte alla. Problem med att det kan ta lång tid innan man får fram en körbar prototyp kvarstår dock (Highsmith J, 2000).

Rapid prototyping

Denna typ av livscykel har under senare år fått mer och mer stöd (Thompson & Wishbow, 1992 samt Gutierrez O, 1989). I korthet går Rapid prototyping ut på att göra så många och korta iterationer som möjligt för att på detta sätt i slutet på varje iteration skapa mervärde för kunden. Detta oftast i form av, som namnet antyder, en körbar prototyp. Denna prototyp behöver dock inte vara ett körbart system, utan skulle kunna vara en "mockup" i form av t.ex en Powerpoint-presentation. Vid slutet av varje iteration säkerställs tillsammans med kunden att systemets utveckling går åt rätt håll. På detta sätt är kunden och den framtida systemägaren hela tiden med under resans gång och har möjlighet att påverka oftare än vid sekventiell eller iterativ utveckling. Denna typ av utveckling får även till följd att kravhantering snarare görs i form av demonstrationer för kunden där frågor som "är det så här ni vill ha det?" ställs för att få bekräftat att utvecklingen av systemet har rätt inriktning. Det finns både fördelar och nackdelar med detta arbetssätt. Fördelarna är bl.a att man inte hinner springa iväg så långt med felaktiga krav utan stoppas av kunden i ett tidigt stadium. En nackdel kan vara att p.g.a dessa snabba iterationer kan dokumentationen bli lidande, man fokuserar all kraft på att utveckla den för iterationen specifika produkten (Thompson & Wishbow,1992).

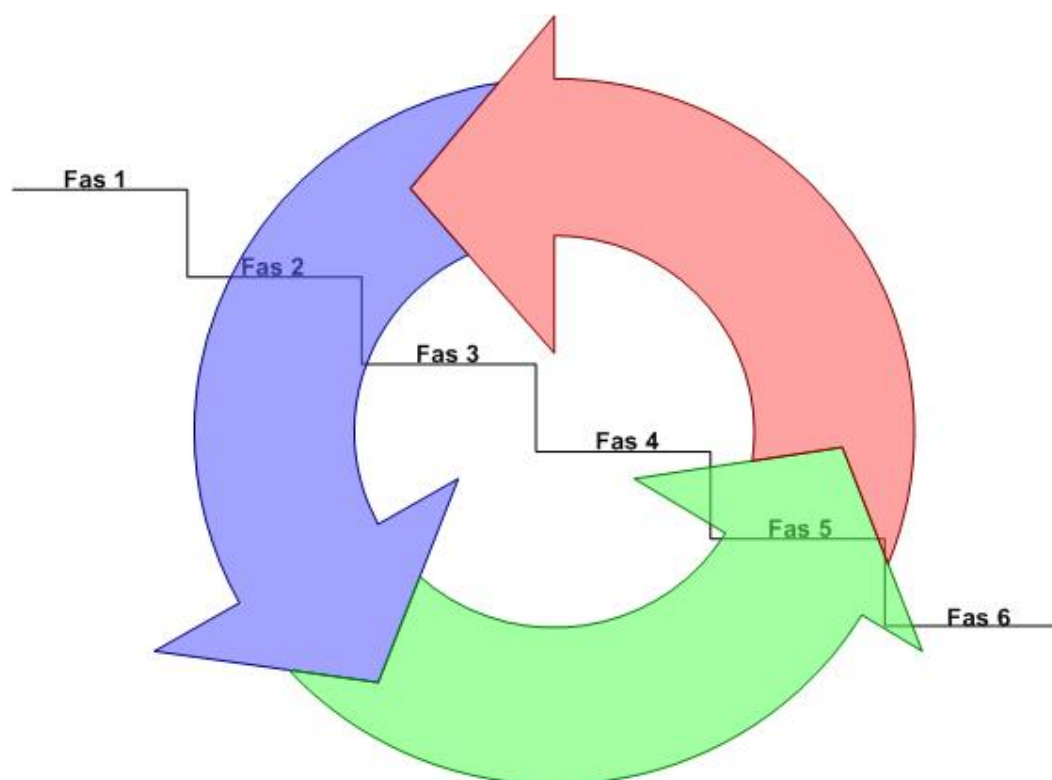


Fig. 3, Rapid prototyping.

Synsätt på metodologianvändning

Metodologier kan och kommer alltid att användas på olika sätt av olika individer och organisationer (Fitzgerald B, 1998), detta är något som inte går att bortse från. Nedan följer ett antal olika synsätt på hur metodologier kan användas, dessa synsätt skall inte ses som absoluta sanningar utan närmast som aspekter av olika filosofier.

Prescriptive

Detta sätt att se på metodologianvändning, som dök upp i slutet på 1970-talet (Avison D. E, 2003), innebär att metodologin är menad att följas strikt från start till slut, i föreskriven ordning. Anhängare till denna typ anser att denna allena metodologityp löser, mer eller mindre, samtliga problem och så länge inte avsteg görs från den utstakade vägen kommer målet att nås på ett effektivt sätt utan att gå miste om för många detaljer. Denna genre benämns även ibland "out-of-the-box" på grund av möjligheten att ta den precis som den är och börja använda direkt utan att göra några anpassningar vare sig m.a.p organisationen eller projektet som metodologin kommer att användas i. Anledningen till att inga anpassningar görs är rädsla för att förlora delar av metodologins funktionalitet (Avison D. E, 2003).

Ett målande citat i linje med detta synsätt:

"Losers consist of unnamed, unspecified, up to the individual, or 'written-but not formalised' types of methodologies."
Zolnowski & Ting, 1982

Contingency

Detta synsätt, eller snarare teori, går ut på att världen ser inte likadan ut överallt och således inte de system kommer att byggas heller. Detta i sin tur gör att det inte finns en enda metodologi som har svaret och lösningen på allting. Man måste kunna ha flera olika verktyg i sin verktygslåda och välja det bästa för den aktuella situationen eller projektet (Avison D. E, 1995, s.103). Detta säger att ju fler metodologier utvecklare kan, desto större är chansen att projektet skall lyckas, förutsatt att utvecklarna väljer den metodologi som är bäst anpassad för att lösa problematiken.

Method engineering

Method engineering är ett synsätt inriktat på att hantera den inneboende inflexibiliteten i en metodologi. Detta uppnås genom att bryta ned metodologin i dess beståndsdelar (se nedanstående beskrivning av en metodologis beståndsdelar), som kan kallas för metodfragment, "method chunks" (Ralyté & Rolland, 2001). Dessa fragment kapslas in, d.v.s de i metodologin föreskrivna kopplingarna mellan fragment bryts upp för att åstadkomma ett resultat där varje fragment är väl definierat och har ett utanför metodologin inneboende värde. Fragmenten kan således paketeras tillsammans med en användarbeskrivning (t.ex ett typfall, där det aktuella fragmentet kan användas). Samtliga fragment kan sedan sparas i ett "repository" vilket skulle kunna beskrivas som en "fragmenttunna". Dessa disparata delar kan sedan "monteras ihop" till en "ny" metodologi, t.ex anpassad för ett enskilt systemutvecklingsprojekt (Funk P. J, 2000). Ofta är det någon människa eller instans av organisationen utsedd till "method engineer" som utför ovanstående arbetsuppgifter. Fragmenttunnan kan vid senare tillfällen fyllas på med andra väldefinierande fragment som inte nödvändigtvis har sitt ursprung i den första dekonstruerade metodologin. Detta får till följd att de nya metodologier som fogas samman av redan befintliga fragment kommer att ha sitt ursprung i flera olika, i sig redan beprövade, metodologier. På detta sätt så växer nya

metodologier fram evolutionsmässigt, där enbart de för uppdraget relevanta fragmenten används.

En vedertagen fördel med "method engineering" är bl.a att då den för tillfället använda metodologin är sammansatt av den aktuella organisationen för det aktuella projektet bidrar detta till att öka acceptansen för den för projektet valda metodologin (Henderson-Sellers B, 2003). En nackdel som ofta nämns i diskussionen kring "method engineering" är att organisationen sällan eller aldrig har tid att vänta på att en skräddarsydd metodologi skall skapas eller resurser för det samma (Fitzgerald B, 2003).

"Hybridmetoden" – Motorola

Detta synsätt förespråkar även det anpassning så som "method engineering" angriper problematiken (Fitzgerald B, 2003). Skillnaden ligger i att anpassningen görs i två steg, på makro- respektive mikronivå. På makronivå hanteras anpassningen av en eller flera, i organisationen ansvariga som i införandeförfarandet ser till att plocka bort och lägga till de delar som denne/dessa finner lämpligt för att metodologin skall passa organisationens verksamhet. På denna nivå sker även kontinuerligt underhåll och uppdatering av metodologin. På en lägre nivå, den så kallade mikronivån, sker anpassning dels i början av varje projekt, där man konstaterar vilka delar som behövs för just detta projekt. Utöver detta har varje enskild medarbetare i viss mån möjlighet att välja de metodologidelar som han eller hon finner lämpliga för att lösa problemet framför sig.

Denna tvåstegraket medför att arbetssättet blir optimerat för organisationens utvecklingsprojekt på makronivå samtidigt som det optimeras på mikronivå inför varje projekt. Utöver detta uppmuntras medarbetarnas kreativitet, eftersom de har möjlighet att använda de lösningsvägar som de själva finner bäst lämpade, förutsatt att de krav som ställs på själva lösningen uppnås (Fitzgerald B, 2003).

Methodologies/Procedures/ Techniques/Tools

Systemutvecklingsmetodologier är enligt Avison och Fitzgerald (Avison & Fitzgerald,1995) uppbyggda av tre huvudkomponenter: procedures, techniques och tools, se figur 4.

Methodologies

Avison och Fitzgerald definierar metodologi på följande sätt:

"... a collection of procedures, techniques, tools and documentation aids which will help the systems developers in their efforts to implement a new information system. A methodology will consist of phases, them-selves consisting of sub-phases which will guide the systems developers in their choice of the techniques that might be appropriate at each state of the project and also help them plan, manage, control and evaluate information systems projects."
(Avison & Fitzgerald,1995, s.10)

Det finns ingen enskild, allomfattande definition av termen metodologi. Ovanstående är ett försök att definiera denna term, som är så generellt att de flesta kan godta detsamma. Dock finns ett otal andra definitioner, mer eller mindre generella. Detta faktum bör beaktas i diskussioner rörande metodologier.

Avison och Fitzgerald (1995) påpekar också att en metodologi för att just vara en metodologi, också ofta har ett filosofiskt perspektiv, detta kan innebära att metodologin fokuserar på t.ex mänskliga aspekter av systemutveckling eller vetenskapliga dito. I de fall där en samling procedures, tools och techniques inte hålls samman av ett filosofiskt perspektiv kallar Avison och Fitzgerald detta för en metod, vilken kan jämföras med ett recept.

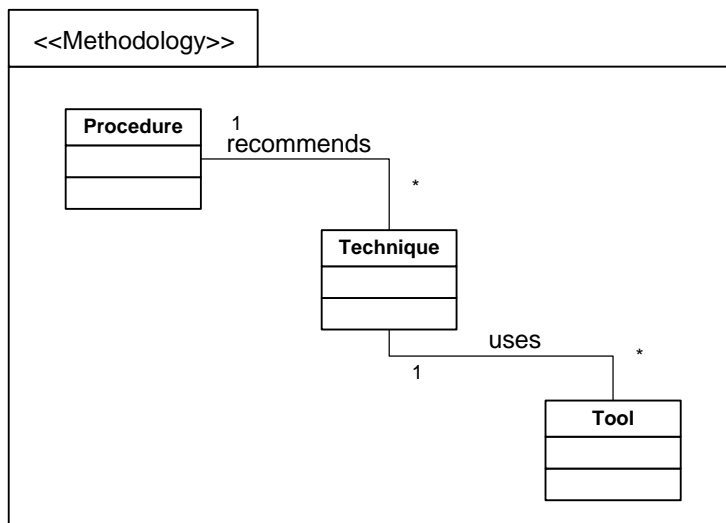


Fig. 4, Procedures, techniques, tools

Procedures

Procedures, som i det närmaste kan översättas med tillvägagångssätt, bestämmer i vilken kronologisk ordning saker och ting sker. Det är dock viktigt att påpeka att procedures inte rör dimensionerna vad och hur utan endast rör sig i dimensionen när. Detta skulle kunna exemplifieras av hur vattensfallsmodellen (Avison & Fitzgerald,1995) beskriver i vilken ordning de olika faserna skall utföras. Analysfasen kommer t.ex före implementeringsfasen och testfasen bör följa implementeringen.

Techniques

Inom procedures återfinner vi olika techniques som talar om vad som skall göras, t.ex hur ett specifikt problem kan lösas. För att återgå till det ovanstående exemplet, vattenfallsmodellen, så skulle framtagandet av en objektmodell i analysfasen vara en technique. Denna beståndsdel behandlar således dimensionen vad. Techniques rekommenderar ofta användandet av specifika verktyg s.k tools.

Tools

För att åter knyta an till exemplet ovan, för att skapa objektmodellen så kan t.ex designspråket UML användas (OMG, 2004). UML spelar här rollen av verktyg (tool) för att underlätta objektmodelleringsprocessen.

Samverkan mellan Procedures, Techniques och Tools

Procedures, techniques och tools behöver inte nödvändigtvis vara beroende av varandra utan kan återfinnas inom många olika metodologier, tools och techniques kan återkomma på flera ställen i olika procedures och kan i stort sett betraktas som självständiga byggstenar. T.ex kan UML användas som tool i flera disparata kontexter men inom metodologins ramar.

Pyramiden

Följande teori är mycket vanligt förekommande inom den företagsekonomiska kontexten, tyvärr är den så vanligt förekommande och så många olika "bastardvarianter" av modellen finns att det är svårt att spåra ursprunget. Den har helt enkelt blivit så pass internaliserad att den är en integrerad del av kontextens kulturella arv.

Enligt organisatoriskt inriktad beslutsteori fattas beslut i organisationer på tre klart urskiljbara plan: det strategiska planet, där beslut som rör organisationens strategi och relationer med omvärlden tas (Hatch M. J, 2002, s.300), det taktiska planet, där beslut som rör hur man på bästa sätt implementerar, m.a.p organisationens resurser och strukturella möjligheter, strategiska beslut, samt det operationella planet, där beslut som rör "producerandet" av resultaten ifrån tagna strategiska och taktiska beslut.

Beslutsfattare på respektive nivå är: strategisk nivå - högsta ledningen, taktisk nivå - mellanchefer, operationell nivå - linjechefer alternativt operatörer.

Strategisk nivå - exempel på beslut: Ledningen bestämmer att hallonmarmelad skall tillverkas, p.g.a det blev en god hallonskörd i år (omvärldsfaktorer), samt att konkurrenten Önös just har lanserat hallonmarmelad som säljer bra (konkurrenter), kunder har via webben uttryckt önskan om hallonmarmelad.

Taktisk nivå - exempel på beslut: Mellancheferna omsätter beslutet att tillverka hallonmarmelad: "Japp, grabbar och tjejer, idag skall vi tillverka hallonmarmelad. Ni två fixar kokningen. Ni två kryddar, ni två fyller burkar."

Operationell nivå - exempel på beslut: Operatören Olle tycker att han gör ett smidigare jobb om han fyller en stor skål med marmelad och sedan från den fyller 8 burkar på en gång, istället för att fylla en burk åt gången.

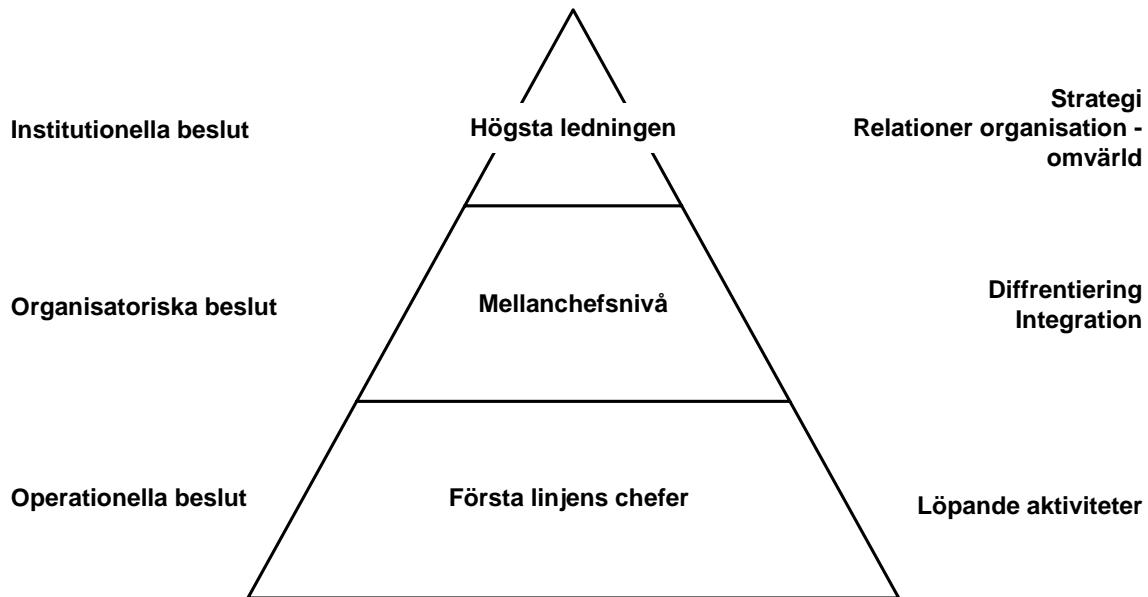


Fig. 5, Pyramiden (Hatch M. J, 2002, s.300)

Perrows teknologitypologi

Att försöka kategorisera olika teknologier är något som organisationsforskare har roat sig med under längre tid. Många olika teorier och modeller har kommit och gått under årens lopp, så som Thompsons typologi (Hatch M. J, 2002, s. 167) och Woodward's typologi (Hatch M. J, 2002, s.165). En man vid namn Charles Perrow är upphovsman till nedanstående modell (Hatch M. J, 2002, s.168) som är en förfining av dom båda tidigare nämnda forskarnas modeller. Perrows menade att det går att dela in typen av arbete i fyra olika genrer, baserade på hur stor variation arbetsuppgifterna har samt vilken möjlighet utförarna av arbetet har att analysera uppgiftens natur.

		Uppgiftsvariation	
		Låg	Hög
Möjlighet att analysera uppgiften	Hög	Rutin	Ingenjörarbete
	Låg	Hantverk	Icke rutin

Fig. 6, Perrows teknologitypologi (Hatch M. J, 2002, s.168)

Dessa fyra kategorier beskrivs enligt Perrows som följande:

- **Rutinarbete**
Denna typ av arbete har låg variation i uppgiften samt kännetecknas av en stor möjlighet att analysera hur arbetet bäst skall utföras. Ett bra exempel på detta är masstillverkande industrier så som bilindustrin
- **Hantverksarbete**
Kännetecknas av både låg variation och låg möjlighet att analysera uppgiften. Här återfinns t.ex byggnadsarbeten, det är låg variation i arbetet men när det väl uppstår problem så måste dessa hanteras och lösas ad hoc.
- **Ingenjörarbete**
Här återfinns både hög variation och hög möjlighet att analysera problemet. Som namnet antyder så är ingenjörarbete ett exempel på detta, god kännedom om problemet möjliggör också hantering av en större uppgiftsvariation.
- **Icke rutinbaserat arbete.**
När det rör sig om stor variation och liten möjlighet att analysera uppgiften rör det sig om ett arbete med ständigt stor osäkerhet. Det kan t.ex vara forskare som ständigt bryter ny mark och därför har begränsade referensramar att dra slutsatser ifrån.

Hatchs femcirkelmodell

Följande modell används i uppsatsen som ett verktyg för att kategorisera anpassningsdrivande faktorer. Meningen är således inte att med hjälp av modellen förklara eller utreda organisatoriska termer. Modellen erbjuder en något mer komplex syn på organisationen, vilket torde hjälpa oss att kategorisera drivande faktorer på ett någorlunda verklighetstroget sätt.

Enligt Mary Jo Hatch (Hatch M. J, 2002, s.34) så kan organisationer uppfattas som bestående av teknologier, fysiska strukturer, sociala strukturer och kulturer, se nedanstående figur. Denna syn på organisationen medför att ingen av de fyra ensamt kan bilda en total förståelse för organisationen utan först tillsammans kan helheten synas. Dessa fyra cirklar är inneslutna i en större cirkel som betecknar omgivningen. Detta för att göra oss påmindna om att precis som dom fyra strukturerna påverkar organisationen så påverkar också omgivningen de fyra strukturerna.

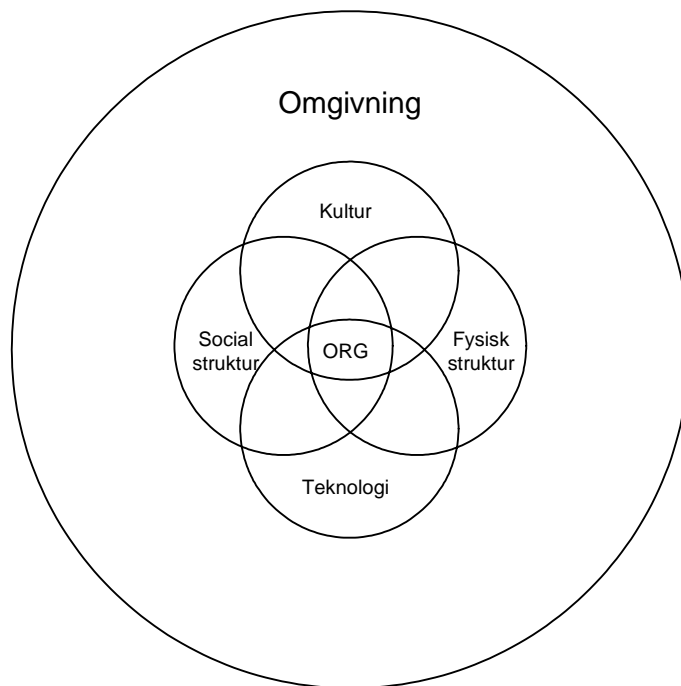


Fig. 7, Hatchs femcirkelmodell (Hatch M. J, 2002, s.34)

- **Kultur**
En amalgamering av förgivettaganden av organisatoriska artefakter som medvetet eller omedvetet växt fram eller skjutits in i organisationen under dess levnadstid.
- **Fysisk struktur**
Rör organisationens geografiska omgivning samt mer fysiska element så som lokaler och byggnader.
- **Social struktur**
Representerar den organisationsform som valts. Exempel på detta kan vara divisionalisering (Mintzberg, H.(1983)).
- **Teknologi**
Utgörs av de produkter, redskap som används i verksamheten, inkluderar också kunskap, s.k ”know-how”.

Egen förklaringsmodell

För att försöka förstå metodologianvändningen i organisationer och ha någonting att hänga upp intervjudiskussioner på, har en modell tagits fram. Modellen skall ses som ett sätt för författarna och kanske också läsarna, att i dagens situation, med metoder/metodologier som spänner över allt fler verksamhetsområden och som ingående delar i allt större och mer komplexa modeller av verkligheten, ha en gemensam bild över hur olika omfattande metoder/metodologier kan samverka och användas.

Modellen har abstraherats för att vara tillräckligt generell. Detta innebär att avsikten med modellen är att den skall kunna appliceras på de flesta situationer inom sitt användningsområde.

Process

Den grövsta granularitetsnivån. Inte nödvändigtvis en systemutvecklingsprocess, utan kan även vara en övergripande projektprocess eller en delprojektprocess. Används företrädesvis när en metod/metodologi innehåller en explicit icke direkt systemutvecklingsanknuten del, t.ex projektstyrning eller när en systemutvecklingsmetod/metodologi ingår som en integrerad del i en större process, t.ex en verksamhetsutvecklingsprocess.

Metodologi

En metod/metodologi kan sägas vara innesluten i en process i det avsett att själva systemutvecklingsarbetet kan existera i ett grövre sammanhang i vilket det sker fler saker. Dessutom kan flera systemutvecklingsmetoder/metodologier innehållas i en process, även när det gäller en eventuellt specifik systemutvecklingsdel. Alltså, olika metoder/metodologier med olika brett spektrum kan utan problem rymmas i en systemutvecklingsprocess.

Aktivitet

Den minsta beståndsdelen i en metod/metodologi. Inom olika metoder/metodologier kan storleken, grovleken samt innehållet skilja sig i stor omfattning. Dock, en aktivitet är alltid en atomär enhet.

Exempel på aktivitet kan vara upprättandet av ett klassdiagram. För att utföra aktiviteter används verktyg (tools) och tekniker (techniques). Olika metoder/metodologier kan innehålla olika antal aktiviteter.

Interaktion med "Pyramiden"

Vår förklaringsmodell har starka kopplingar till den tidigare förklarade "pyramiden". Vi upplever att beslut som rör övergripande process i dom allra flesta fallen tas på en strategisk nivå. Dessa beslut kan röra t.ex val av metoder/metodologier. På den strategiska nivån är det vanligt med beslut gällande individuella projekt. Exempel på detta skulle kunna vara beslut om att inte använda sig av en viss disciplin i metodologin RUP för innevarande systemutvecklingsuppdrag. Den operativa nivån kännetecknas av beslut som tas av den operativa personalen, d.v.s beslut som rör den dagliga verksamheten i ett systemutvecklingsprojekt. Exempel kan vara beslut om hur ett klassdiagram skall upprättas. Ovanstående exempel är givetvis arketyper och det förekommer naturligtvis undantag beroende på projekts uformning och omfattning.

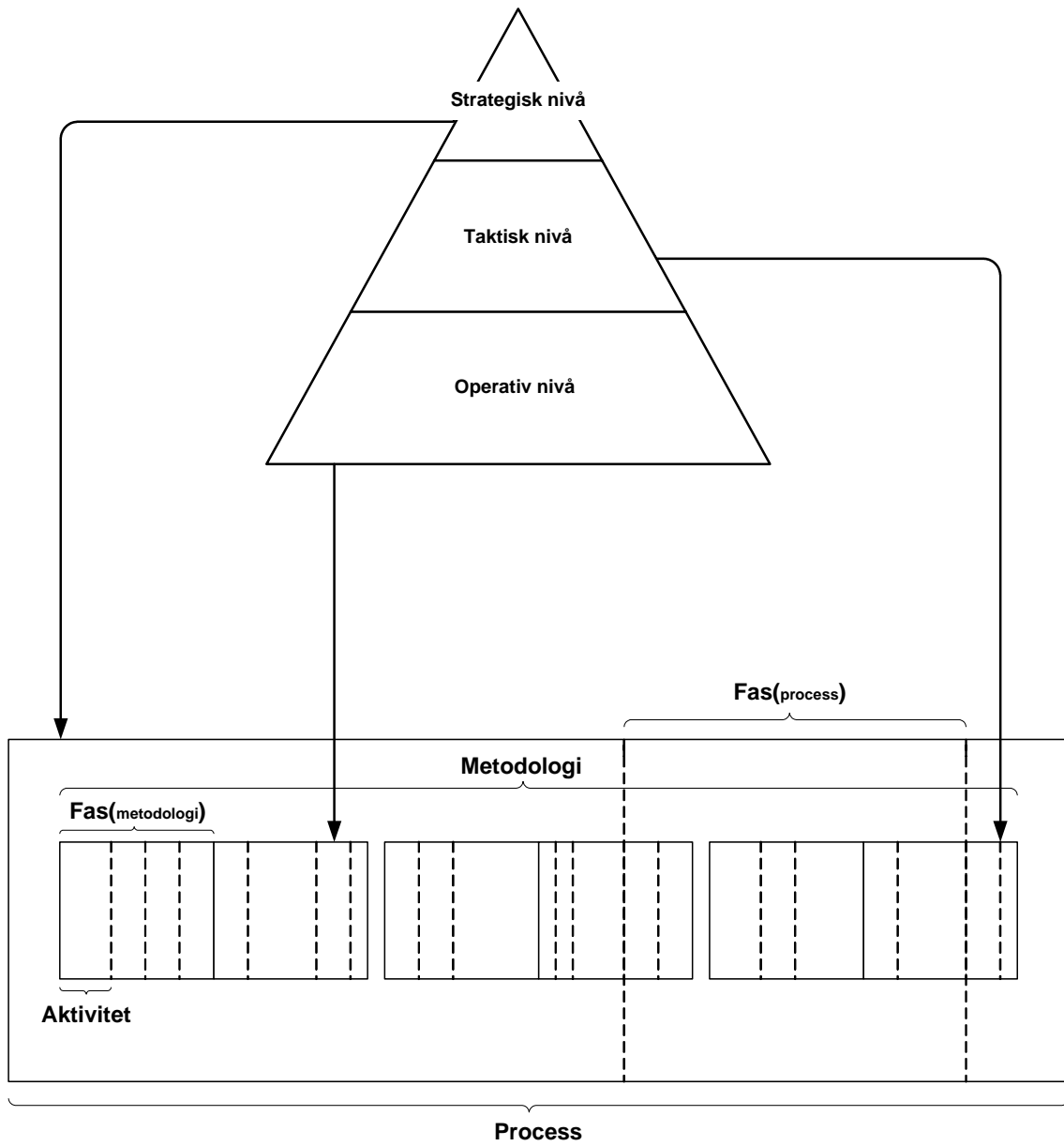


Fig. 8, Egen förklaringsmodell

Egen modell för anpassningstypologier

Författarna har omarbetat Perrows teknologitypologi för att kunna använda den för att kategorisera organisationer vars kärnverksamhet är baserad på någon form av systemutvecklingsuppdrag.

Axeln uppdragens variationsgrad (y), skall tolkas som ett mått på bredden av de uppdrag som organisationen tar/kan ta. Som exempel kan rena konsultföretag, som tar uppdrag så varierade som nyutveckling av mindre webb-projekt till förvaltning av stordatorbaserade s.k. legacy-system, vilket skulle visas i modellen som hög variationsgrad.

Behov av anpassningsflexibilitet (x), å andra sidan, skall tolkas som ett relativt mått på hur mycket eller hur önskvärt/viktigt det är att kunna anpassa sin valda systemutvecklingsmetodologi till rådande förhållanden/uppdrag. Kanske något självklart torde det framgå att de företag som baserar sin verksamhet på en egenutvecklad produkt, med de begränsningar det innebär i bredd och uppdragsvariation, inte har speciellt stora behov eller starka önsningar om att kunna anpassa sin metodologi på projektbasis.

Företag med egenutvecklad IT-produkt

Med denna rubrik avses företag som har utvecklat/utvecklar och säljer sin produkt. Ofta görs mindre anpassningar av produkten till specifika kunders förhållanden. Dock är produkten mer eller mindre standardiserad till sitt utförande. Lokala kundanpassningar kan t.ex göras med hjälp av s.k applikationskonsulter under implementationsfas på plats hos kund. Exempel kan vara affärssystemleverantörer.

IT-konsultföretag utan specifika allianser

Kan också kallas "rena konsulter", d.v.s företaget tar uppdrag som endast är limiterade av bransch-gränser. Företaget kan t.ex ta uppdrag som utveckling av mindre webb-projekt, men också nyutveckling av större administrativa system, t.ex lönesystem och personaladministreringssystem.

IT-konsultföretag med specifika allianser

Med detta menas företag som, förutom att fungera ungefär som 'rena konsulter', också av olika anledningar har ingått strategiska allianser med mjukvaruföretag. Dessa allianser syftar främst till att snabba upp processen för företaget att vid uppdragsstart undersöka vilka mjukvaror som skall användas hos kund. Det finns också en klar marknadsföringsaspekt, i vilken företaget drar nytta av sina allianspartners goda rykte (och förhoppningsvis också tvärtom). Exempel här kan vara ett företag som genom sina allianser kan erbjuda sin kund ett brett spektrum av standardlösningar från välkända aktörer på marknaden i ett snävt tidsperspektiv, för att sedan göra kundspecifika anpassningar om så behövs.

Spelföretag

Med ovanstående menas dataspelsutvecklingsföretag. Dessa företag kan utveckla ett antal spel under t.ex ett års tid. Ur ett branshperspektiv ingår samtliga deras utvecklingsuppdrag i en ganska snäv domän, men dessa uppdrag kan vara väldigt olika till sin art, beroende på speltyp, omfattning o.s.v. Traditionellt sett har dataspelsbranschens användande och anpassning av systemutvecklingsmetodologier varit liten eller icke-existerande, men med spelens ökande komplexitet, bl.a p.g.a kundernas krav på verklighetstrogenhet har behovet av formalisering och stöd i utvecklingsprocessen ökat markant. M.a.p ett typiskt spelutvecklingsprojekts geografiska spridning i utvecklingsstadiet, bör behovet av en gemensam utvecklingsmodell öka väsentligt.

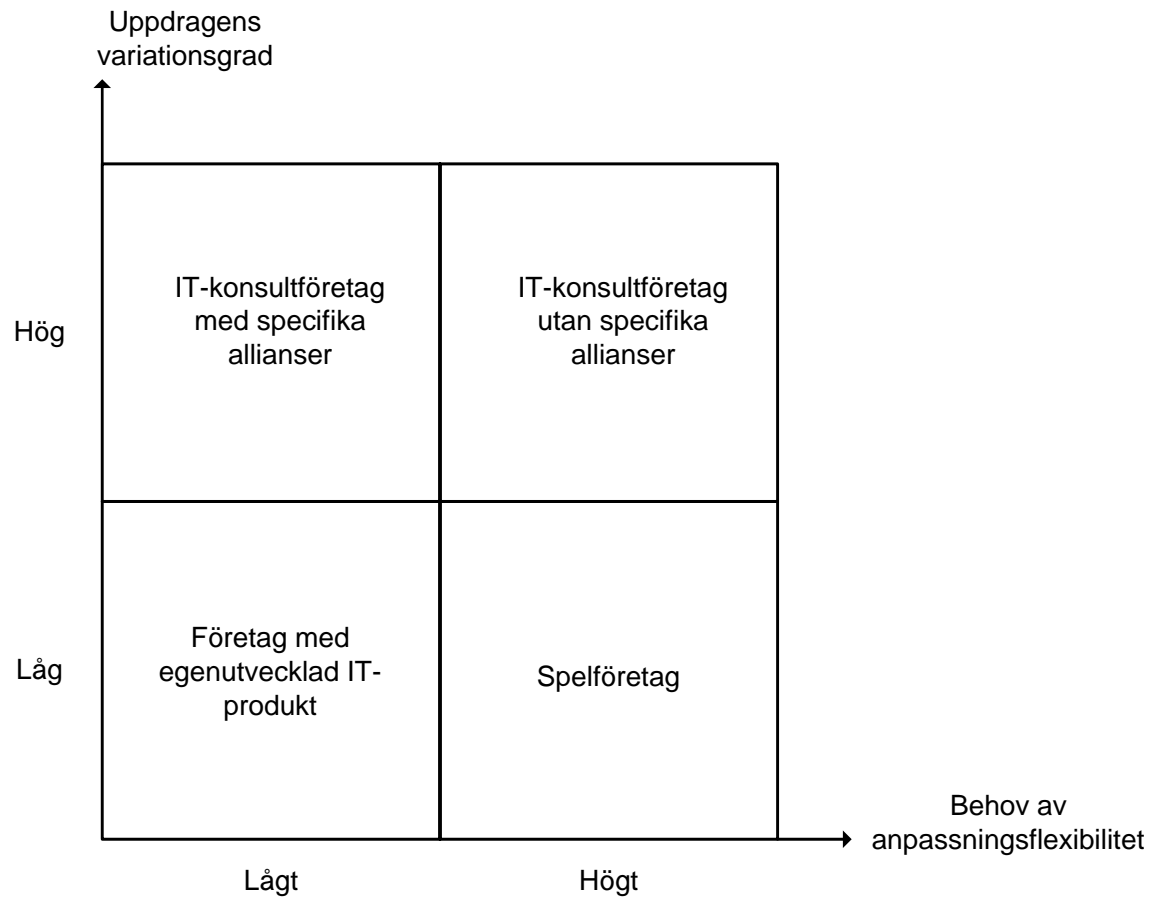


Fig. 9, Egen anpassningstypologi

Empiri

Under processen med att fastställa företag att intervjua så hade vi följande tankegång: vår avsikt var att få möjlighet att intervjua företag med så olik verksamhet som möjligt för att om möjligt kunna dra slutsatser kring deras behov av systemutvecklingsmetodologier. Den ideala situationen hade varit att vi fått till stånd intervjuer med: tillverkande företag, renodlade konsulter samt spelutvecklare. Ingen av de tillfrågade spelutvecklingsbolagen hade dock tid och/eller möjlighet att ta emot oss för en intervju. Detta har fått viss återverkan på studien (se kapitlet slutsats, avsnitten reliabilitet och validitet).

Tilläggsas bör även att vi som sagt inte hade för avsikt att diskutera någon specifik metodologi, denna typ av diskussion gick tyvärr dock inte att undkomma då de intervjuade företagens referensram var just den av organisationen använda metodologi. Av denna anledning och enbart denna anledning förekommer diskussioner kring metodologin RUP återkommande i intervjumaterialet.

Vi vill redan här klargöra att vi inte har för avsikt att återberätta hela intervjumaterialet utan istället arbeta med intressanta citat. Den intresserade läsaren återfinner intervjumaterialet i sin helhet under bilaga 2 och framåt.

Vår analys av intervjumaterialet kommer således inte här utan vi syftar enbart till att här återberätta de intressanta fakta som har uppkommit under intervjuerna med de fem företagen.

IFS

IFS (IFS, 2004) är ett företag som är aktivt inom affärssystemutveckling, de utvecklar och marknadsför sin egen produkt, IFS Applications. IFS Applications är ett moduluppbyggt affärssystem, med ett antal grundmoduler, till detta kommer skräddarsydda kundanpassade lösningsmoduler. Produkten riktas främst emot följande sju segment:

- Automotive
- High tech and medical device
- Industrial manufacturing
- Process industries
- Service and facilities management
- Utility and telecom
- Aviation, rail and defense

Förutom utveckling och design förekommer även konsulting enbart riktad mot det egna systemet. IFS ser sig själva som ett transnationellt företag, då de finns representerade med 79 kontor i 45 länder. Huvudkontoret i Göteborg har cirka 300 anställda, företaget som helhet har dock cirka 2680 och omsätter årligen 319 miljoner dollar. Större delen av produktutvecklingen sker på Research & Development-avdelningen (R&D) på Göteborgskontoret och utvecklingsimplementeringen sker huvudsakligen i Sri Lanka och Tucson, Arizona, USA. Detta medför att det inte är ovanligt med projekt med mer än 100 involverade utvecklare som löper över mer än ett år, där individerna dessutom är fysiskt placerade i flera olika världsdelar.

Beskrivning av metodologianvändande

IFS använder sig av en organiskt framväxt egen metodologi som kallas för Paketmetodiken, denna metodik går i korthet ut på kortare utvecklingscykler eller paket om fyra veckor som påbörjas med en analysfas och avslutas med en testfas, inte helt olik vattenfallsmodellen enligt den intervjuade. Det som tål att noteras är dock att dessa testfaser för att optimera effektiviteten löper något omlott för att på så sätt programmerarna i Sri Lanka skall kunna rätta till föregående paket samtidigt som nästa paket förbereds av designers i Sverige. Se nedanstående figur.

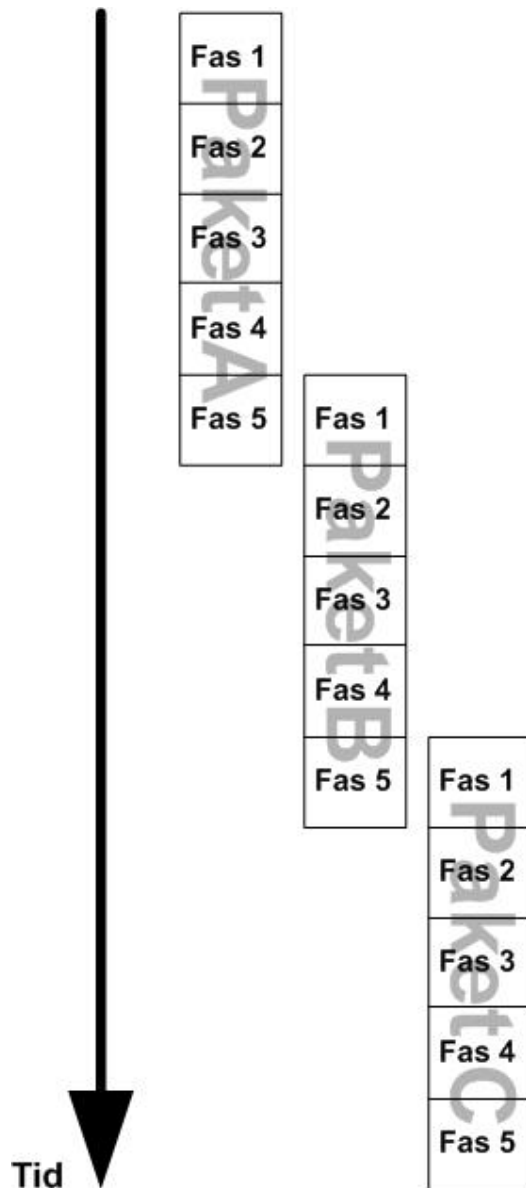


Fig. 10, IFS paketmetodologi

På detta sätt används utvecklingsresurserna i Sri Lanka och Sverige på bästa möjliga sätt. Dödtiden då utvecklarna inte har något att utveckla och designers inte har något att designa blir minimal utav den enkla anledningen att paketen löper omlott. Paketmetodiken används oavsett vilka av dom sju affärssegmenten som utvecklingen sker mot, eller som den intervjuade på IFS sa: ”*Ett krav är ett krav*”. Dessa krav måste hanteras oavsett om det är processindustrin eller telekomsektorn som är kravställaren. Den intervjuade tillade dock att när det är försvarsindustrin som äger kraven så kan det ställas högre krav på sekretess och informationsspridningsmodeller, i övrigt är allt lika. Paketmetodiken är något som har vuxit fram mer eller mindre evolutionsmässigt på företaget och sedan formaliserats av en arbetsgrupp. Tidigare användes något som man skulle kunna kalla för ”ad hoc”-metodik, problemet löstes på det sätt som befanns mest lämpligt. Efter att IFS vuxit fort och mycket, strax före 2000, krävdes dock ett gemensamt arbetssätt, detta för att förenkla flyttning och sammanslagning av resurser mellan olika projekt. Detta nya arbetssätt är rollbaserat och inte namnbaserat.

Paketmetodikerna bygger förutom på sättet att leverera paket, på ett antal estimeringsmodeller och påläggsfaktorer. Detta för att kunna skapa tidsuppskattningar så nära det verkliga utfallet som möjligt. För att personalen skall kunna ta till sig detta arbetssätt sker utbildning inom koncernen kontinuerligt. Det finns även en mycket detaljerad databaslösning på nätet som går under namnet "Metodwebben", denna weblösning fungerar som en kunskapsbank där projektdeltagare kan söka hjälp om de är osäkra på hur metodologin skall användas och vilka leverabler som aktuell aktivitet skall mynna ut i.

Metodologianpassning hos IFS

Paketmetodikerna togs fram med hjälp av ett antal olika metod-fragment som fanns och användes inom företaget. Dessa sattes sedan ihop och anpassades för att passa produktutvecklingen inom de olika bolagen i IFS-koncernen. Eventuella projekthanpassningar kunde även ske i början av varje projekt.

"...när vi jobbar med projekt så är det i princip alltid samma faser i ett projekt när man installerar ett affärssystem."

IFS

Ett utvecklingsteam har tämligen stor frihet i hur de löser utvecklingsuppdrag, dock, uppdragens resultat är hårt specificerade. Om enskilda projektmedlemmar finner brister i metoden kan dessa flaggas upp och efter en formaliseringsprocess återfinnas i metodologin. Det är således en levande metodologi som utvärderas och uppdateras kontinuerligt.

Det är en viss skillnad på konsultverksamheten och R&D. Den största skillnaden återfinns dock hos Research där ett antal olika metodologier används eftersom en standardiserad metodologi för utveckling över flera kontinenter lätt hämmar forskningen. Den blir för stor helt enkelt.

"Och hur jag skall ta mig igenom dom stegen, dom aktiviteterna som finns men det är normativt, det beskriver alltså vad är normalfallet och det innebär att i alla dom fallen som ehh...där verkligheten skiljer sig från modellen så måste man tänka själv och man måste göra dom förändringarna som krävs för just den situationen som uppstår så att man får inte följa en metod slaviskt, men man får heller inte vara för slapp i sitt utnyttjande av metoden för då är det väldigt stor risk att man tappar greppet om projektet."

IFS

ADB-kontoret

ADB-kontoret (ADB-kontoret, 2004) är en förvaltning inom Göteborgs stad med cirka 300 anställda som arbetar med alltifrån drift av befintliga system till nyutveckling. Av dessa 300 arbetar ungefär 90 med systemutveckling. Att organisationen är en förvaltning inom Göteborgs stad innebär att organisationen är styrd av kommunallagen och, som komunal förvaltning, befinner sig i en något ovanlig konkurrenssituation. ADB-kontoret har inte möjlighet att som privata företag bedriva aktiv utåtriktad marknadsföring av sina produkter. Dock, försäljning av produkter till kunder utanför de kommunala förvaltningarna är möjlig genom så kallad merförsäljning. Merförsäljning innebär att ADB-kontoret får lov att sälja ett redan befintligt system förutsatt att det inte kräver externa resurser i form av t.ex marknadsföring och/eller vidareutveckling/anpassning av produkten. Detta har i sin tur fått till följd att organisationen känner sig utsatta för konkurrens på ojämna villkor, eftersom Göteborgs stad är fria att förhandla med vilka parter som helst, men ADB-kontoret endast får bedriva verksamhet mot Göteborgs stad.

På grund av att organisationen har till uppgift att samla upp uppdrag inom Göteborgs stad så ser uppdragen väldigt olika ut. Det kan röra sig om alltifrån ekonomisystem till understödjande system riktade mot socialtjänsten. Detta får i sin tur till följd att det är en stor diversitet i de olika teknologier som används inom organisationen.

Beskrivning av metodologianvändande

ADB-kontoret har gått från att ha olika utvecklingsmetodologier på varje enskild avdelning, och mer eller mindre vattentäta skott mellan dessa avdelningar, till att införa RUP i organisationen.

Införandet av RUP började för ungefär två år sedan och är ännu inte helt klart. Till RUP har man valt att addera sin egen projektstyrningsmodell samt sin egenutvecklade verksamhetsmodellering. Detta val har gjorts på grund av att man upplevde att dessa var mer heltäckande än motsvarande discipliner i RUP, samt i fallet med verksamhetsmodelleringen, att kunderna var så pass bekanta med denna typ av modellering att man skulle förlora för mycket på att frångå det nuvarande arbetssättet. Vad gäller systemutvecklingen har man dock valt att helt låta RUP ta hand om detta. I organisationen finns även ett ”RUP-team” tillgängligt som kan bistå projektorganisationen vid projektstart med insikter i hur RUP bäst skall användas.

Metodologianpassning hos ADB-kontoret

ADB-kontoret valde att införa RUP under en omorganisation och ett av kriterierna för val av ny metodologi var just att den skulle vara anpassningsbar till en sådan verksamhetsbredd som organisationen har. En annan faktor vid val av metodologi, och det som gjorde att valet föll på att köpa en redan befintlig sådan, var enligt de intervjuade, just kostnaden det medför att underhålla och uppdatera en egen metodologi. Man ansåg sig inte ha resurser nog för att göra detta på ett tillfredsställande sätt.

Redan vid införandet av RUP plockades delar som ansågs överflödiga bort och de egna metodologierna, som projektstyrning och verksamhetsmodellering plockades in. ADB-kontoret har även utökat RUPs ”best practices” med sina egna standarder för t.ex use-case. Inför varje nytt projekt görs ett så kallat ”development case”, där man ser över projektet och anpassar metodologin efter de förutsättningar som ges. Vid varje avslutat projekt ses den för projektet lokala anpassningen över för att ta med sig eventuella erfarenheter in i den standardiserade metodologin.

WM-data

Företaget grundades 1969 och har sedan dess varit aktivt inom IT-konsultbranschen. WM-data (WM-data, 2004) har länge varit kända för att ha ett brett spektrum av tjänster, år 2000 renodlades dock organisationen och WM-datas nya profil togs fram. Företaget står nu på tre ben: systemutveckling, leverans av applikationer och IT-drift. Omsättningen uppgår till 8 miljarder och företaget sysselsätter 8000 personer. Företaget är och kommer, enligt uppgift, att förbli ett nordiskt företag som riktar sig mot de 60 största företagen i Norden, de så kallade "top 60"-företagen. WM-data riktar sig således inte till slutkonsumenten utan enbart till större företag och organisationer i Norden. Affärsidén bygger på att bistå kunden under hela processen, att vara med under kravfångst, design, arkitektur, implementation, drift, förvaltning och support. Vad gäller systemutveckling så används något som WM-data själva kallar för modelldriven systemutveckling. Detta går i korthet ut på att så mycket kod som möjligt skall kunna genereras utifrån de modeller som har byggts upp under designfasen. Detta möjliggörs genom kopplingar mellan och s.k plugins till ritprogram som Microsoft Visio och Rationals verktyg och systemutvecklingsverktyg som Microsoft Visual Studio. Detta frigör tid för utvecklare som i sin tur kan lägga mer energi på affärslogiken i systemen.

Beskrivning av metodologianvändande

VM-datas huvudsakliga metodologi är RUP kombinerat med designspråket UML.

Kvalitetssäkring sker genom att konsulterna RUP-certifieras av IBM.

Till RUP adderas ett antal inom WM-data befintliga metodologier så som:

- **Ratten**
En projektstyrningsmetodologi som täcker in det som vi i vår modell kallar för process. Ratten är uppbyggd kring milstolpar för att säkerställa processens flöde. I övrigt bygger metodologin på interaktion med kunden och checklistor.
- **PRM**
Project Risk Management, som akronymen skall utläsas, jobbar som namnet antyder med riskeliminering och uppföljning, att rätt prestanda och korrekt design uppnås.
- **Accept**
Används för att modellera processer i ett väldigt tidigt stadium, till och med före use-case skapas. Denna metodik skall öka förståelsen för vilka dataflöden och casedata som det tilltänkta systemet ställer krav på.
- **Delta**
Används för att skapa en interaktionsdesign mellan människa och dator, en storyboard.

Utöver detta nämnde den intervjuade att WM-data även har en egen systemutvecklingsmetodologi vid namn SUMO. Erfarna konsulter har även kunskaper inom de stora kundernas egna utvecklingsmetodologier och processer, bland de stora kunderna märks bland andra AB Volvo och Ericsson.

WM-data använder sig utöver detta för något som dom har valt att kalla för modelldriven utveckling. Denna typ av utveckling bygger på en koppling mellan det program där man ritat sitt klassdiagram och den utvecklingsmiljö som man skriver koden i. Denna egentillverkade koppling medför att man kan få databasstrukturer, get-, och set-metoder samt den vanligaste affärslogiken genererad direkt från Visios eller Rationals vertyg rakt in i Visual Studio.

Lösningen är programberoende tack vare att modellerna beskrivs via XML.

Till detta kommer WM-datas Business Object Framework som är en egenutvecklad API.

För att allt detta skall kunna hanteras av medarbetarna på ett smidigt sätt, finns en databas tillgänglig på företagets intranät som behandlar återvinning. I denna weblösning kan osäkra användare plocka fram gamla RUP-artefakter, smarta kodlösningar m.m.

Metodologianpassning hos WM-data

RUP, som är den huvudsakliga metodologin, anpassades en gång vid införandet för att stämma överens med verksamheten och sedan en gång i början av varje projekt, ett så kallat development case. Valet av RUP var naturligt eftersom kunderna tror på det. Den anpassning som WM-data valt att göra förutom att addera egna metodologier till RUP (se ovan) är som vi redan nämnt att använda sig av vad som inom företaget kallas modelldriven utveckling. Detta för att förkorta ledtiderna i början på projekten och för att på detta sätt bli mer konkurrenskraftiga.

"...i stora projekt kan man sitta i månader innan man har koll på, hur gör vi det? Varför gör vi det? Vad skall vi göra? Och det kostar stålar, ni vet det kostar grymt mycket stålar."

WM-data

RUP används för att det är ett "internationellt språk" och för att det har färdiga mallar, dokument med rubriker att fylla i. Även detta ökar accelerationshastigheten i början på projektet och man anser sig komma upp i "styr fart" snabbare. WM-data anser även att en av RUPs stora fördelar är att den besätter projekt via roller. Ett 30-tal roller finns definierade och funktionsbeskrivna. En annan fördel med RUP, enligt WM-data, är att den har så kallade "best practices" där man kan fråga Rational hur man skall lösa en viss typ av problem. Varje projekt får ett paket med WM-data Development Framework som innehåller kodkonventioner, guidelines och arkitekturlösningar som måste användas. Detta blir projektets "lager". På detta sätt strömlinjeformas tänkandet så att samtliga medarbetare får samma bild av vad t.ex en treskiktslösning är.

"Det gäller att kvalitetssäkra saker som namnsättning, designpatterns etc. det här kör vi, vi kör ingenting annat och gör vi något annat så får vi fasen motivera varför vi gör det. Vi har inte råd att låta folk sitta och hacka, det går inte."

WM-data

"Den här kör ni efter, punkt slut, det finns inga alternativ. Bara kör."

WM-data

Kunskapsdatabaser, mönster för hur man kommer undan prestandafällor med mera, har evolverat. Artefakter från tidigare RUP-projekt som kan användas som hjälpmedel är ett tydligt exempel på detta.

Att WM-data jobbar med implementatörer i Polen och Indien, så kallad offshoreutveckling, ställer även det höga krav på en tillförlitlig metodologi. WM-data har löst detta med hårda kopplingar från modellen in i utvecklingsmiljön. Arkitekturen stannar i Sverige men implementeringen sker där det är billigast.

"Det är inte så lång tid sedan det verkligen var så att man ville ha en duktig VB-utvecklare. Men den frågan finns inte längre. Det är intressant, för...man vill ha nåt mer. Man vill ha folk som förstår disciplinerna."

WM-data

Intentia

Intentia (Intentia, 2004) utvecklar, anpassar och säljer sin egen produkt, affärssystemet Movex. Den konsultverksamhet som sker inom bolaget sker enbart på uppdrag baserade på den egna produkten. Intentia finns representerade i ett 40-tal länder med totalt 3000 anställda. Göteborgskontoret har cirka 350 anställda. Den totala omsättningen uppgår till 3.5 miljarder kronor. Systemutvecklingen, som har sitt centrum i Sverige, sker till största delen på en Java-plattform. Denna plattform har även en tillhörande egenutvecklade systemutvecklingsmiljö. Utvecklingsmiljön har tagits fram av Intentias Research and Development-avdelning för att nå upp till de krav som Intentia ställer. Intentia utbildar sina kunder och anställda via Intentia University som är ett renodlat utbildningsföretag. Företaget har två versionssläpp per år, dessa datum är heliga och får inte flyttas, om inte allt hinner bli klart till deadline, skär man i den ursprungliga kravs-specifikationen.

Beskrivning av metodologianvändande

Intentias arbetssätt styrs övergripande av att dom är ISO9000-certifierade. Denna certifiering beskriver processarbetssättet. Så länge denna process följs är den individuella utvecklaren fri att skapa en lösning som uppfyller kraven på det sätt som denne finner lämpligt. Istället för att slaviskt följa någon metodologi har Intentia valt att efterfölja ett antal kod- och dokumentations-standarder. Dessa standarder har evolverats fram över tiden. Anledningen till att detta går att genomföra utan att alla gör helt som de själva vill beror enligt de intervjuade på att standarderna är inkorporerade i företagets egenutvecklade utvecklingsmiljö. Detta gör att utvecklingsmiljön tillhandahåller ett antal sätt att hantera valuta, språk, tabeller, utseende, utskrifter o.s.v. På detta sätt ersätter utvecklingsverktyget till viss del en systemutvecklingsmetodologi. Utöver detta kommer ett eget ramverk i Java som även det strömlinjeformar organisationens sätt att utveckla system.

Metodologianpassning hos Intentia

I samband med en omorganisation passade ledningen på att kvalitetsstämpla utvecklingsarbetet genom att certifiera sig med hjälp av ISO-organisationen. Vad gäller systemutvecklingsmetodologier på lägre nivå än processövergripande har även de vuxit fram över åren. Nya saker som tas in i standardsättet kommer ofta från prototypprojekt där utvecklarna är lite friare att arbeta som de själva vill och kanske testat något nytt som senare inkorporeras i standarden.

”Ja, det är ju att någon har hört talas om eller känner till eller sett eller det här kan man använda, det här är bra att använda här och så provar man det, då.”

Intentia

En större anpassning som skedde var när valet av programmeringsspråk byttes från RPG till Java. Eftersom Java var objektorienterat ställde detta nya krav på metodologin.

På den operativa nivån är utvecklarna väldigt fria att välja den metodologi som de finner bäst. Anledningen till att detta fungerar är enligt de intervjuade att Intentias egenutvecklade utvecklingsmiljö strömlinjeformar arbetssättet. Utvecklarna kan inte annat än följa utvecklingsmiljön och den tillhörande plattformen. Som vi redan nämnt har den standarder för hur saker som grafiska användargränssnitt, valuta, språk och utskrifter skall hanteras.

”Jag vet inte om man kan ge en definition på det så, det har ju...genom åren har det ju.. det är ju teorier egentligen bara. Och dom är mer eller mindre bra, mer eller mindre löser alla problem i världen, men dom fungerar ju aldrig, teorin är ju aldrig likadan som verkligheten.”
Intentia

Accenture

Accenture (Accenture, 2004) är ett renodlat internationellt konsultbolag med 83.000 anställda och finns representerade i 47 länder. Denna konsultjätte har en omsättning på 11.8 miljarder amerikanska dollar eller i svenska kronor, med dagens penningvärde, 91.3 miljarder kronor. I Sverige finns cirka 600 medarbetare och på kontoret i Göteborg finns 90 av dessa. Accenture börsintroducerades 2001 på New York-börsen och var innan dess partnerägt, d.v.s ett antal individer som nått statusen ”partner” i bolaget hade ett vinstdelningsprogram. För att kunna vara med och slåss med de andra jättarna som IBM behövdes dock större finansiell uthållighet och därav den publika introduktionen på börsen. Organisationen är uppbyggd kring roller och avancemang, du går in som analytiker och klättrar sedan i grader. Detta kan liknas vid militära organisationer. Ju högre upp du kommer i organisationen, ju mer förväntas du också sälja in nya projekt till kunder i ditt kontaktnät.

Företagets kunskapsbas är bred, tyngdpunkten ligger dock runt IT- och management-konsultverksamhet. Följande discipliner har Accenture i sin portfölj:

- Strategy & business architecture
- Customer relationship management
- Supply chain management
- Human performance
- Finance & performance management
- Technology research & innovation
- Solutions engineering, outsourcing
- Technology solutions

Accenture har, som redan påpekats, en mycket bred plattform att stå på och åtnjuter ett gott anseende i branschen när det gäller att arbeta med välrenommerade företag.

En av deras strategier är allianser med andra företag. Dagens IT-projekt är enligt företaget större och kräver mer än vad en ensam aktör kan leverera. Accenture har därför allierat sig med framstående produktföretag bland dessa märks t.ex Microsoft, SAP, Oracle och HP

Beskrivning av metodologianvändande

Accenture använder en egenutvecklad metodologi som har vuxit fram under många år. Visserligen har den genomgått en rad namnbyten men idén är fortfarande den samma vilket den intervjuade påpekade. Metodologin går nu under namnet ”Accenture Delivery Suite” och bygger på ett flertal olika mallar och templat. Företaget har även en egen ”instans” av sin metodologi för varje typ av teknologi som de allierat sig med. Det finns t.ex en metodologiinstans för SAP-integration, en för Peoplesoft och en för Microsoftteknologier o.s.v. Att ha en stor egen metodologi kräver stor stödorganisation vars arbete går ut på att underhålla och uppdatera metodologin. Mycket på grund av att Accenture är ett så pass stort företag och ofta tar stora projekt som löper över flera länder, ställs höga krav på att allt skall se likadant ut vartän på jorden du arbetar. Detta märks t.ex vad gäller alltifrån tidsredovisning till systemutveckling, något som framkom tydligt under intervjun.

Inom företaget finns också kunskap om och erfarenhet av de stora kundernas valda metodologier, t.ex General Motors egen metodologi och Volvos användning av RUP. Allt detta understöds av en metodweb med färdiga mallar, templat och givetvis beskrivning av de olika metodologiinstanserna.

Metodologianpassning hos Accenture

När Accenture gick ifrån "Method one" på 80-talet till BIM, Business Integration Model, på 90-talet, gjordes detta på grund av att kunderna efterfrågade något mer än bara ett färdigt system. Kunden ville även ha strategiutveckling och drift av system. När det sedan var dags att gå ifrån BIM till den nuvarande metodologin, "Accenture Delivery Suite" gjordes det på grund av att allianserna hade vuxit sig så pass starka att ett mönster kunde skönjas i projekten. Det finns, enligt uppgift, mycket god intern kunskap om de allierades produkter, vilket leder fram till anpassade metodologiinstanser för just dessa produkter. Accenture har t.ex så pass många integrationsprojekt med SAP att man såg en effektiviseringsmöjlighet i att skapa en metodologi för just SAP-integration. SAP är inget specialfall, utan det gäller likväl Microsoft-arkitekturer och uppsättning av större Oracle-system. En annan faktor som har lett fram till anpassningar i Accentures metodologi, är i enlighet med intervjun, att företaget är multinationellt och bedriver offshoroutveckling. Detta ställer helt andra krav än om utvecklingen sker i samma land och/eller i länder med högt utbildade utvecklare med förståelse för analys och design.

"Det fanns ingen beskrivning för hur man hanterade 100 programmerare i Indien...så att omvärlden förändrar nog eller tvingar fram en förändring av vår organisation eller av vår affärsmodell."

Accenture

Det framkom även under intervjun att ju mer globalt ett projekt är, desto viktigare blir det att inte göra avsteg från metodologin.

Analys och diskussion

Som den uppmärksamme läsaren fortfarande kommer ihåg, är vår frågeställning som följer:

Hur anpassar organisationer systemutvecklingsmetodologier?

Vilka faktorer driver anpassningen av systemutvecklingsmetodologier?

Här nedan presenteras vår analys av det empiriska materialet.

Indelning av företag

I enlighet med den teori som benämns ”egen anpassningstypologi” har vi sett ett mönster bland de intervjuade företagen. Företag som Intentia och IFS, som utvecklar en egen produkt, har ett behov av anpassningsbarhet i metoder/metodologier som är betydligt mindre än vad företag som WM-data och ADB-kontoret har. Företag som knyter sina underleverantörer eller mjukvarutillverkare till sig har även de behov en särskild typ av anpassning. Detta mynnar ut i att organisationer av denna typ skapar skräddarsydda instanser av sin systemutvecklingsmetodologi för att maximera nyttan av densamma mot varje enskild produkt. Vi tror att, även om detta är mer eller mindre spekulationer, att de två spelutvecklingsföretagen, som vi tyvärr ej fick möjlighet att intervjua, borde ha ett stort behov av att kunna anpassa metoden/metodologin, speciellt med tanke på den ibland väldigt stora skillnaden mellan spel och spel (jämför racingspel, där banan och kringmiljöns utformning och verklighetstrogenhet är extremt viktig, och t.ex realtidsstrategispel där enhetsbalans och ”artificiell intelligens” har stor betydelse). Vi kan tyvärr inte annat än att spekulera i denna typ av systemutvecklingsföretags behov av systemutvecklingsmetodologier.

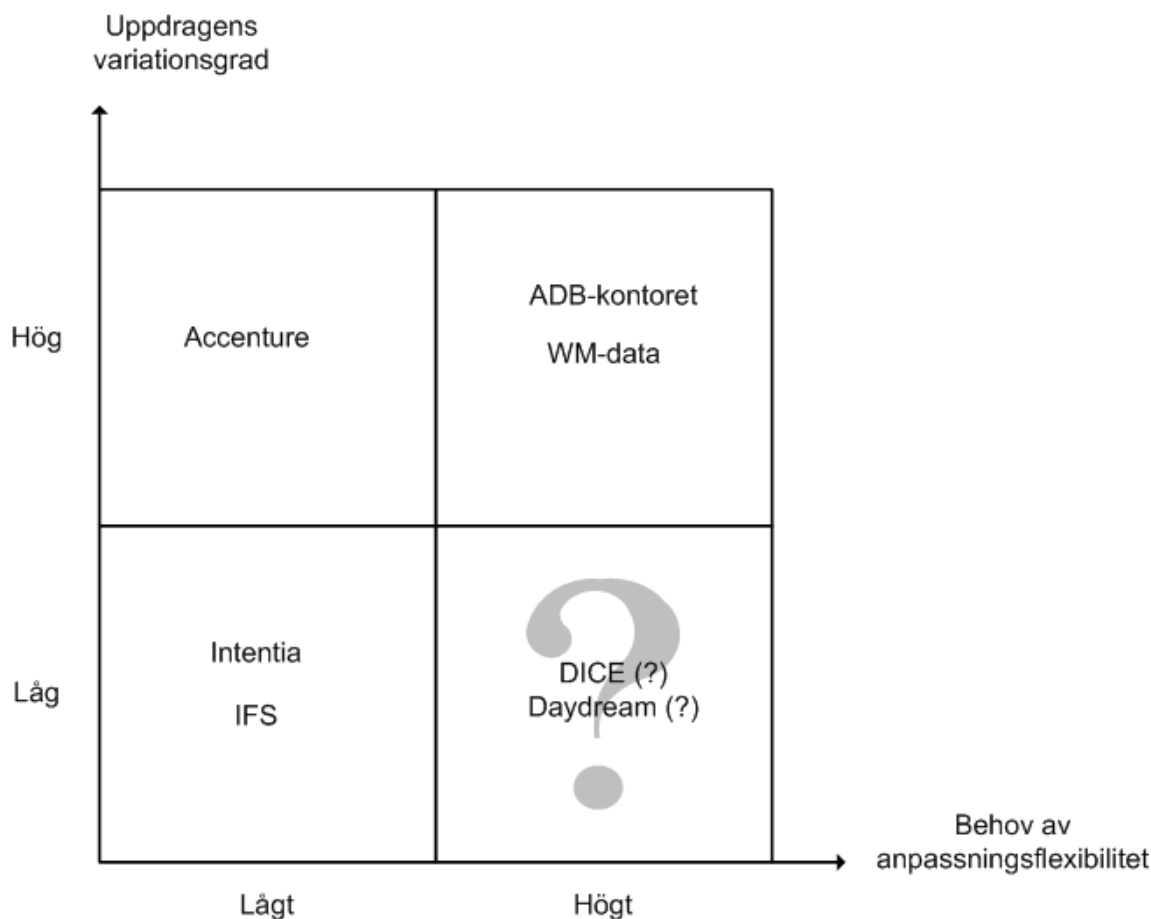


Fig. 11, Indelning av företag

Analys av empiriskt material m.a.p användning och anpassning av metodologier

Vid en genomgång av vårt empiriska material har vi försökt att extrahera samtliga anpassningar rörande metoder/metodologier. Dessa anpassningar kan kategoriseras med hjälp av tidigare elaborerad teori, benämnd "pyramiden". På strategisk nivå rör det sig om anpassningar av engångskaraktär. Det kan vara anpassning till verksamheten eller till organisationens specifika produktval. Man kan även urskilja anpassning till systemutvecklingsspråk och tillhörande verktyg. Det sker även anpassningar för att väva samman systemutvecklingsmetodologin med organisationens övriga metodologier, så som t.ex projektstyrning. Anpassningar på denna nivå förblir statiska så länge som organisationen och/eller verksamheten inte genomgår några större förändringar.

På den taktiska nivån återfinns anpassningar som rör varje individuellt pågående projekt. Anpassningen sker således i början av projektet och lever enbart under projektiden. Här sker anpassningen mot de förutsättningar som projektet i fråga har, vilket ger möjlighet till väldigt stora avvikelser i metodologianpassning från projekt till projekt.

På den operativa nivån är det anpassningar som görs av det individuella teamet eller den individuella projektdeltagaren under projektets gång för att möta dagliga förändringar och utmaningar som uppstår.

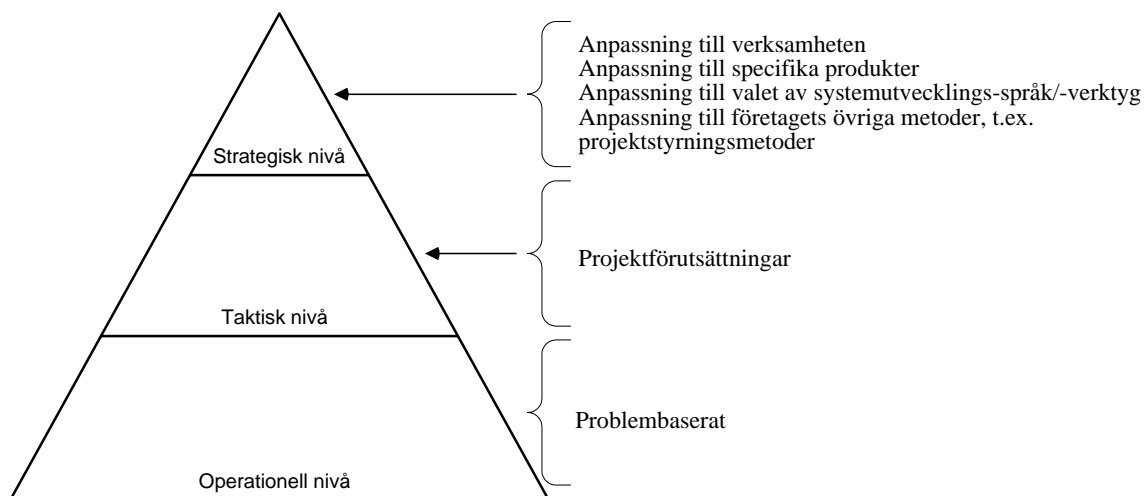


Fig. 12, Uppdelning av anpassningar

Verktøget blir metodologin

Ett genomgående tema i det empiriska materialet är att metodologianvändning i mångt och mycket bottnar i att styra systemutvecklingsverksamheten i rätt riktning samt att få mätbara punkter med vars hjälp återrapportering till ledning och projektägare kan möjliggöras. Speciellt gäller detta företag som jobbar med utveckling och marknadsföring av sin egen produkt, så som IFS och Intentionia. I dessa företag blir metoden/metodologin mer av ett styrverktyg än en utvecklingshjälp. Dessa företag har i tämligen stor utsträckning försökt att integrera metodologin i sina utvecklingsverktyg, vilket framkommer tydligt i följande del av en intervju:

Intenia: ”Vi kör en egen utvecklingsmiljö, som innehåller dom grejer som behövs... det är ganska standardiserat, vi jobbar med språk till exempel. Det skall fungera för alla världens språk och så vidare, det finns krav på kontanter, meddelanden, allt sådant där och det stödjer den... hyfsat lätt att översätta och så där. Tabeller, det är också standardiserat det skall vara så här och vi har verktyg som tar fram det utseendet.”

Björn Olsson: ”Det har tagit ett tag att ta fram den utvecklingsmiljön...”

Intenia: ”Ja...men den ändrar sig och den lever hela tiden och sen är det ju nya delar så... Vi har...i vår visuella del verktyg för det.. standardiserat på ett visst sätt. Ett visuellt lager som har ett visst utseende och det ställer vissa krav på att man skall göra på ett visst sätt och då har vi verktyg som stödjer detta och man kan inte göra något annorlunda...”

Ovanstående förfarande får ett antal konsekvenser varav den ena är att metodologins synlighet minskas inom företaget, ”man ser inte skogen för alla träd”. Den andra är att metodologin inte behöver läras ut till nyanställda i sin helhet. Verktöget lärs ut istället och därmed också metodologin. Detta kan uttryckas som att ”verktöget är metodologin”.

En kontrast till ovanstående kan exemplifieras med hjälp av WM-data. WM-data har med sitt angreppssätt, modelldriven utveckling, gått så långt som möjligt i samma riktning som Intenia, men med tanke på att WM-data inte har en egen produkt utan är ett renodlat konsultbolag är uppdragen väldigt varierande. Detta får till följd att det inte är möjligt att gå lika långt i införandet av metodologierna i verktygen. WM-data har istället hittat beröringspunkterna mellan analys/design och programmering och gjort en integreringslösning. Ett exempel på detta är integrering av UML-bibliotek i utvecklingsverktöget. Denna lösning förhindrar att något annat än UML-notation används vid verksamhetsmodellering. En ytterligare konsekvens blir att ledtiderna förkortas då modellen går rakt in i utvecklingsverktöget.

Kartläggning av drivande faktorer

De anpassningsdrivande faktorer som har extraherats ur det empiriska materialet har vi valt att kategorisera enligt Hatchs femcirkelmodell (se teoriavsnittet). Femcirkelmodellen har valts för att om möjligt öka förståelsen för ur vilken organisatorisk kontext dessa anpassningsdrivande faktorer springer. Samtidigt vill vi dock påpeka att vi inte ser det som vår uppgift att diskutera t.ex företagskulturer som enskilda begrepp. Denna diskussion lämnar vi med varm hand över till renodlade organisationsforskare. Vi använder således femcirkelmodellen som ett verktyg för att i första hand sortera in faktorerna och skapa överblickbarhet. Denna modell valdes då vi fann att pyramidmodellen inte gav tillräckligt med stöd för att på ett tillfredställande sätt dela in faktorerna.

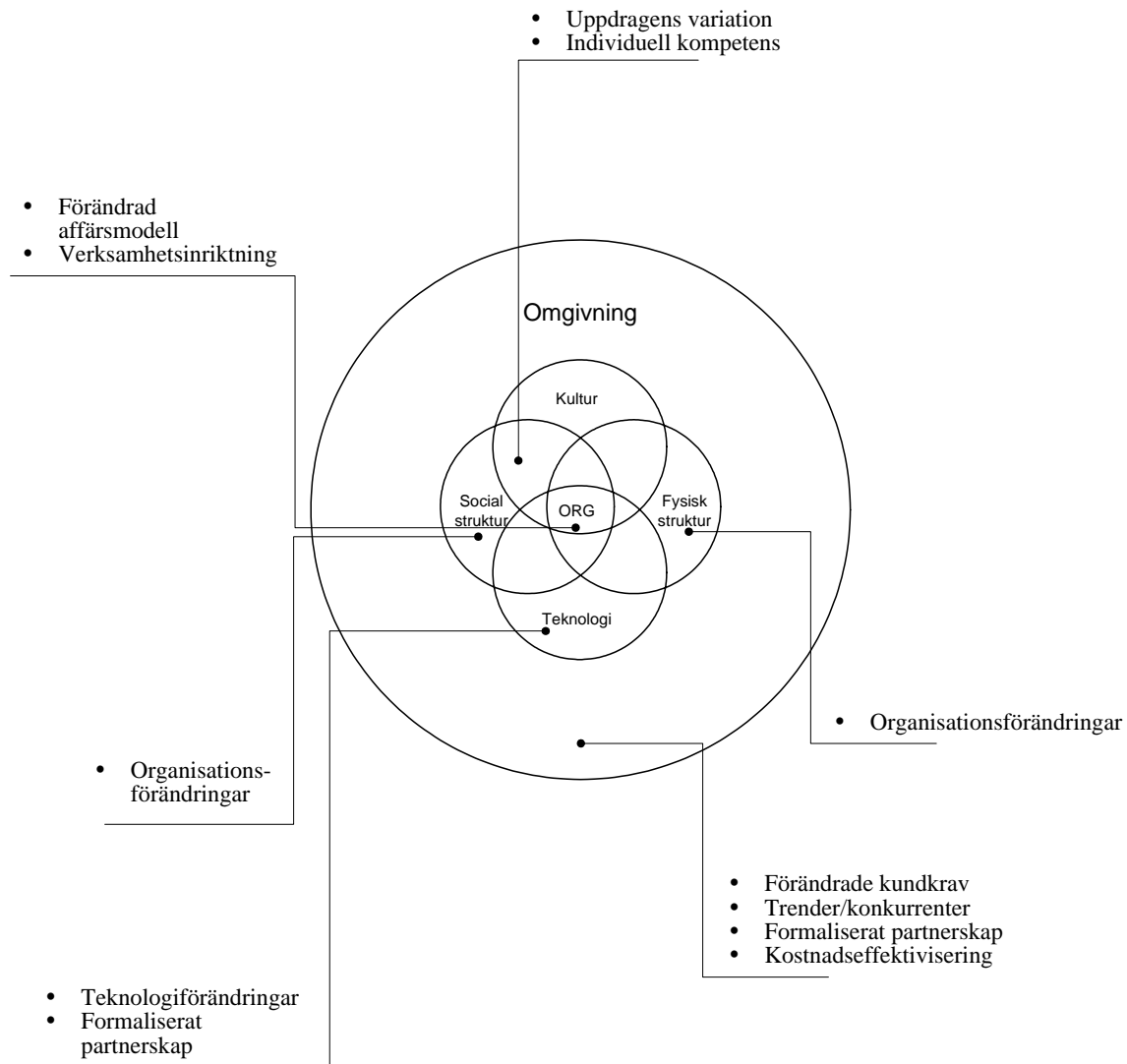


Fig. 13, Kategorisering av drivande faktorer

Förklaring av drivande faktorer

- Förändrad affärsmodell**
 Detta är besläktat med faktorn benämnd ”verksamhetsinriktning”, skillnaden ligger i att denna faktor avser evolutionära förändringar av flergångskaraktär. Ett exempel på detta är när en organisation utökar sin verksamhet från att ha haft enbart konsultverksamhet inriktad mot systemutveckling till att även omfatta verksamhetsutveckling.
- Uppdragets variation**
 Verksamhetens variation har naturligtvis bäring på hur stora anpassningsmöjligheter som behövs inom en metodologi. Rendodlade konsultbolag som t.ex WM-data ställer mycket högre krav på flexibiliteten hos en metodologi än vad t.ex ett produktutvecklande företag som IFS gör. Självklart är det så att det faktum att man en vecka kan ha ett uppdrag att utveckla en mindre webbösning och en annan vecka (eller...ja...veckor, snarare) ha ett uppdrag att utveckla ett företagsövergripande

lönesystem gör att fler och större anpassningar av metod/metodologi måste göras.

”Och det kan vara verksamhetsprojekt eller ett infrastrukturprojekt, nu skall vi stoppa in Windows Server 2003 på Borealis, vad gör vi? Jo det skiljer ju sig kraftigt från ett utvecklingsprojekt men i båda fallen behöver man en metodik.”

WM-data

”...vi sysslar med allting, från serverdrift, webhotell, övervakning, systemutveckling, de tidiga faserna i verksamhetsutveckling, kravhantering, stöd vid upphandling av it-system, genom hela spektret.”

ADB-kontoret

- **Förändrade kundkrav**

Majoriteten av de intervjuade företagen anser att förändrade kundkrav är en betydelsefull drivande faktor för anpassning, vilket tydligt illustreras av nedanstående citat.

”Man kan väldigt grovt generalisera, och säga såhär: förr...för en 10, 15, 20, 30 år sen, så fick WM-data frågor av den här typen: vi vill ha en duktig cobol-programmerare, vi vill ha en väldigt duktig VB-utvecklare. Vi får inte dom frågorna idag. Dom frågorna finns inte idag. Det är ingen som vill ha en duktig VB-utvecklare. Helt out alltså. Det är ju rätt konstigt. Det är inte så lång tid sedan det verkligen var så att man ville ha en duktig VB-utvecklare. Men den frågan finns inte längre. Det är intressant, för...man vill ha nåt mer. Man vill ha folk som förstår disciplinerna[RUPs discipliner]”

WM-data

”...det här kanske är en av anledningarna till att utvecklingen...systemutvecklingsmetoderna ändrar sig i det långa perspektivet, att...om vi på 80-talet var en systemleverantör, som levererade ett till ICA, som var en stor kund på 80-talet, ehhh...så har det gått...ja...kunderna frågar efter större och större åtaganden i tid, man vill inte bara ha en komponent eller ett system eller någonting utan man vill att vi utvecklar, levererar och driftsätter någonting, men även sköter driften och det sista steget är ju nu outsourcing där dom till och med vill att vi tar över delar av deras personal för att ja... driva det här i ett antal år framåt.”

Accenture

- **Teknologiförändringar**

Revolutionära teknologiförändringar och paradigmskiften i systemutvecklingstänkande är, föga överraskande, faktorer i metod/metodologianpassning. Det kan verka självklart att en övergång från funktionsbaserat tänkande till objektorienterat tänkande kräver anpassningar av systemutvecklingsmetodologier, men det bör snarare vara byten av metoder/metodologier (vilket i sig är en form av anpassning) som är aktuella, då en metod/metodologi som är sprungen ur ett funktionsbaserat tänkande skulle vara tämligen meningslös i ett objektorienterat paradigm.

”...nu är det Java, men innan var det ju RPG bara och det ser annorlunda ut och då fick också metoden som vi höll på med vara lite annorlunda.”

Intentia

- **Trender/konkurrenter**

Trenders inverkan på val av metodologi skall inte underskattas. Inte heller konkurrenternas göranden och låtanden avseende metodologianvändning kan avskrivas som faktor vid metodologival. Dessa två tillsynes tämligen triviala faktorer inverkar på val av metod/metodologi i det avseendet att en organisation är rädd för att inte få de (faktiska eller inte) konkurrensfördelar som konkurrenterna (kanske eller kanske inte) upplever att de får med sin valda metodologi. Konkurrenternas val av metodologi (och för den delen trender) kan också vara betydelsefull i den meningen att man ser sina konkurrenter som potentiella samarbetspartners i större projekt, med plats för fler konsulter/systemleverantörer och då upplever att det kan finnas en fördel i att ”prata samma språk”, d.v.s använda sig av samma (eller liknande) metod/metodologi.

Ytterligare en anledning att ta trender på allvar visas i nedanstående citat, som sammanfattat kan yttryckas som att de inom flest organisationer använda metoderna/metodologierna har ett försteg vid metodologival just tack vara att de är mest använda, ett slags ”standard”, med väl utbyggda möjligheter både avseende utbildning och förförståelse.

”...då var det just att RUP är ju en välkänd metod då vet jag inte ens om man hade börjat prata lite om XP möjligtvis men dom var ju inte lika kända som dom är nu i alla fall utan det var ju RUP som var den största metoden inom systemutveckling och att det fanns färdiga utbildningspaket inom alla olika discipliner inom RUP och det finns konsulter om man behöver konsult hjälp så är det lätt att få konsult hjälp som vet vad RUP är och kan jobba enligt RUP och kan falla in i vårt naturliga arbetssätt och likadant vid nyrekryteringar att när man rekryterar att man kan säga det att en merit är om du kan RUP så får man in kunnig personal med en gång.”

ADB-kontoret

- **Organisationsförändringar**

Vid ändring av en organisations interna processer och dess uppbyggnad är det inte ovanligt att antingen konformera metodologi efter organisationen eller organisationen efter metodologin. Val av metodologi utgår inte sällan från en omorganisationssituation. Översyn av befintliga metoder/metodologier kan också göras i samband med organisationsförändringar. En något annorlunda typ av organisationsförändringar är uppköp/inkorporeringar. Det är inte ovanligt med uppköpssituationer där den uppköpta organisationen inte har något större val än att använda sig av den uppköpande organisationens befintliga metodologi, snarast utan hänsyn till befintlig metodologi i den uppköpta organisationen.

” Man kan väl säga att från början när vi gjorde vår organisationsförändring, definierade man ju upp våra processer i organisationen, inte bara systemutveckling, utan alla, och så gjorde man organisationen lite utifrån det, då. ’Vad har vi för processer på ADB-kontoret?’ och då var systemutveckling en av dem, av delprocesserna, och det var ju det som gjorde att vi skulle hitta en metod som skulle bli vårt gemensamma arbetssätt för den här processen. Man kan säga att till viss del var det ju processerna som utgjorde underlaget för valet av metod, då.”

ADB-kontoret

” Dom får ju...där har vi ju metoden Implex som är...den finns där den är, den skall dom följa. Sen kan det ju ta en liten stund innan dom kan den, kan följa den. Sen ställer vi ju oftast, jag vet inte, ställer vi ju upp med folk från Sverige som hjälper till från början, som jobbar där och då styr man och hjälper till att komma in i det.”
Intentia

- **Individuell kompetens**

Detta är en anpassningsdrivande faktor, förutsatt att det finns tillräcklig frihetsgrad för individuella utvecklare/projektdeltagare. Ju mer detaljerat en metod/metodologi styr händelseförloppet i en systemutvecklingsprocess, desto mindre utrymme finns det för anpassning utgående från individuell kompetens. Om tillräcklig frihetsgrad ges inom metoden/metodologin finns det stort utrymme för situationsanpassningar av metodologin utgående från individers samlade kompetens och erfarenheter. En hämsko kan vara att i stor utsträckning använda sig av ”verktyget är metodologin” (se tidigare avsnitt i detta kapitel), då det blir näst intill omöjligt att göra anpassningar för den individuella utvecklaren.

”Det är ungefär som... jag brukar använda försvaret som ett exempel där. Ryska försvaret är ju totalt toppstyrt, så skjuter man chefen så vet inte truppen vad dom skall göra, då går dom hem i allmänhet. Medans svenska försvaret fungerar mer så att varje grupp är målstyrd, så dör chefen så kan fortfarande gruppen lösa uppgiften. Så att är det ett hinder på vägen så vet man ändå att man skall ta kullen t.ex, då springer man till kullen och tar den. Det är inte värre än så och det är liksom på ungefär samma ledarskapsmetodiker som paketmetodiken är uppbyggd då. Att folk skall ha förståelse för vad dom skall göra men ha frihet att ta sig fram till lösningen på ett optimalt sätt och så jobbar vi hellre med review då.”
IFS

- **Formaliserat partnerskap**

I de fall en organisation väljer att ingå formaliserat partnerskap med t.ex mjukvaruleverantörer kan detta få konsekvenser för den inom organisationen använda metoden/metodologin. Det finns då utrymme att anpassa den befintliga metodologin tämligen hårt gentemot mjukvaruleverantörens produkt/er. Detta behöver inte innebära att den ursprungliga, mer generella metodologimplementationen ”försvinner”. Resultatet kan ses som en ”instansiering” av metodologin som är mer specifikt styrd av partners mjukvara och som används i en specifik situation. Om ett uppdrag är av en art där en partners mjukvara inte tilltalar/passar kunden används den generella instansen av metodologin.

”Allianser för att projekten blir så stora och kompletta, man frågar efter kompletta lösningar och vi kan inte äga alla utan det blir dom här allianserna att det bildas grupperingar...”
Accenture

”Den stora ändringen nu, från “BIM” till “Accenture Delivery Suite” är att den består av olika moduler och där vill man då ha...vissa moduler är förberedda för dom här standardverktygen, så det finns till exempel en SAP-modul - “så här implementerar vi ett SAP-projekt, såhär implementerar vi olika Microsoft-grejer”, det går mot mer...man skall på nåt sätt tvingas välja några partners för att kunna leverera nånting snabbt, till priset av att vi inte längre kan kalla oss oberoende konsulter.”

Accenture

- **Verksamhetsinriktning**

Beroende på om organisationen i fråga bedriver utveckling av en standardprodukt eller är ett renodlat konsultföretag så ger detta olika behov av metodologianpassning. Ett företag som har en hög variationsgrad i sina uppdrag har större behov av en metodologi som kan anpassas inför varje uppdrag. Ett företag som bedriver utveckling av en standardprodukt har inte det behovet utan ser istället möjligheten att anpassa metodologin och finslipa den i detalj för att passa utvecklingen av en enda produkt.

- **Kostnadseffektivisering**

Kostnadseffektivisering ger naturligtvis återverkningar även på metod/metodologi. Ett aktuellt exempel är, enligt nedanstående citat, offshoreutveckling. Om en metod/metodologi innehåller en implementationsdel kommer denna att påverkas om organisationen väljer att använda sig av offshoreutveckling och anpassning kommer att behövas. Likaså är kostnader för att underhålla en egenutvecklad metodologi en faktor i en metodologivalssituation. Detta p.g.a att det kostar stora pengar att ständigt hålla den egna metodologin uppdaterad, något som mindre organisationer har svårt att mäta med. Krav på kortande av ledtider kan också ge effekter för anpassning av metod/metodologi, detta syns t.ex i WM-datas ”modelldriven utveckling”.

”Det fanns ingen beskrivning om hur man hanterade 100 programmerare i Indien. Så att omvärlden förändrar nog eller tvingar fram en förändring av vår organisation eller av vår affärsmodell.”

Accenture

”Där hade vi en ganska utvecklad egen metod då att välja på, eller någon befintlig på marknaden. Då valde vi bort att ha en egenutvecklad metod för att vi såg att det kräver en ganska stor organisation för att hålla en metod som man själv har utvecklat vid liv. All dokumentation, utbildning och att hålla den up to date o.s.v. Det är nog för stort för ett företag som ADB-kontoret att hålla det här igång och hålla kvaliteten på det.”

ADB-kontoret

Slutsats

Hur anpassar organisationer systemutvecklingsmetodologier?

Som intervjumaterialet indikerat, har organisationer med olika inriktning olika behov av anpassning av systemutvecklingsmetodologier. Dessa behov kan grovt delas in i följande kategorier:

- Företag med en egenutvecklad produkt (IFS, Intention) tenderar att skraddarsy sin valda systemutvecklingsmetodologi för att passa förvaltningen av sin produkt. Denna anpassning är av engångskaraktär, med påföljande, mycket mindre, anpassningar under produktens livstid.
- Renodlade konsultföretag (WM-data, ADB-kontoret) har ett mycket större behov av anpassning på projektbasis än företag som utvecklar och förvaltar en egen produkt. Detta ger i förlängningen större fokus på metodologier som ramverk. Anpassningen sker ofta efter den s.k hybridmetoden (se teorikapitlet), där en större, verksamhetsinriktad anpassning kompletteras med anpassningar på projektbasis.
- Konsultföretag med strategiska allianser (Accenture) instansierar sin generella utvecklingsmetodologi för att passa sina allianspartners befintliga mjukvaror. Detta innebär inte att den generella utvecklingsmetodologin inte används men man försöker minimera dess användning då man upplever en stor fördel m.a.p snabbhet och ekonomi i implementationen när mjukvaruinstanser av metodologin kan användas.

Vilka faktorer driver anpassningen av systemutvecklingsmetodologier?

Följande faktorer har identifierats och presenteras i bokstavsordning nedan. För en djupare beskrivning och förklaring av nedanstående faktorer, se diskussionsavsnittet.

- Formaliserat partnerskap
- Förändrad affärsmodell
- Förändrade kundkrav
- Individuell kompetens
- Kostnadseffektivisering
- Organisationsförändringar
- Teknologiförändringar
- Trender/konkurrenter
- Uppdragets variation
- Verksamhetsinriktning

Reliabilitet och validitet

Reliabilitet är ett mått på hur tillförlitligt det resultat som studien utmynnat i är.

Validitet svarar istället på frågan: Har vi mätt rätt objekt? Nedan presenteras faktorer som vi anser har spelat en aktiv roll m.a.p reliabilitet och validitet.

Reliabilitet

Då det empiriska materialet baseras på intervjuer med endast fem företag begränsas naturligtvis möjligheten att dra generella slutsatser om branschen som helhet. Dock, bland dessa fem företag tycker vi att vi har fått en tämligen god spridning vad gäller verksamhetsinriktning.

Vidare har datainsamlingen begränsats till semistrukturerade intervjuer. Deltagande observationer vilket potentiellt hade kunnat öka reliabiliteten har inte använts av tidsskäl. Med hjälp av dessa observationer hade vi kunnat få bekräftat att metodologierna verkligen användes på det sätt som de intervjuade angav.

Validitet

Intervjudeltagarna har haft olika bakgrund och olika arbetsuppgifter inom sina respektive organisationer. Detta innebär att den mentala distansen till organisationens valda metodologi har varierat. Intervjuade som har liten mental distans till metodologin tenderar att ha mer detaljrik kunskap om den samma. Detta avstånd behöver dock inte vara någon nackdel då detta snarare representerar den verkliga användningen i organisationen.

Förslag till vidare studier

Spelföretag

Det hade naturligtvis varit intressant att få möjlighet till att undersöka spelutvecklingsbolagens metodologianvändande och behov av anpassning, något som så vitt vi vet till dags dato är tämligen outforskat. Utvecklingen av spel ställer allt högre krav på företagen som utvecklar dem och det är inte ovanligt idag att ett antal olika företag är involverade i ett och samma spelprojekt som sträcker sig över två års tid och involverar 50-talet människor, detta enligt diskussioner som pågått dom senaste åren i branschtidningen Game Developer Magazine (GDMAG,1998-2000). Detta ställer naturligtvis krav på metodologier, något som spelbranschen har varit någorlunda befriad ifrån tidigare.

Verktygen är metodologin

Detta fenomen, att inkorporera metodologin i verktygen och på så sätt "tvinga" användaren av verktyget att följa metodologin, är något vi har påträffat under intervjuerna med de företag som saluför och utvecklar en egen produkt (IFS, Intentia).Åtminstone två frågeställningar borde vara intressanta att forska vidare på:

- Vilka konsekvenser får ovanstående på användares uppfattning av metodologin (metodologins synlighet)?
- Vilka konsekvenser får ovanstående på utbildning av utvecklare hos berörda organisationer?

Ordlista

Accept

Metodologi hos WM-data för processmodellering som utmynnar i dataflöden och case-data.

Agile methods

En metodologityp som är gjord för att anpassas.

API

Application Programming Interface. Ett programmeringsgränssnitt mot redan definierad funktionalitet.

BPR

Bussiness Process Reengineering. Att på ett revolutionerade sätt förändra affärsprocesser.

Contingency

Contingencyteorin utgår från att det inte finns en allomfattande lösning som löser alla problem. Problemen ser olika ut, så också lösningarna.

Delta

Metodologi hos WM-data för att skapa en interaktionsdesign mellan människa och dator.

Development case

Den anpassning som görs inom RUP inför varje projekt.

DSDM

Dynamic Systems Development Method. Ett systemutvecklingsramverk som skall vara dynamiskt till sin uppbyggnad.

GUI

Graphical User Interface. Det grafiska användargränssnittet.

Hybridmetoden

Det av författarna benämnda sättet att anpassa i två steg, på strategisk nivå samt på taktisk nivå.

Implex

Intentias metodologi för utrullning av affärssystemet Movex hos kund.

ISO9000

En processtandardiseringscertifiering utfärdad av organisationen, International Organization for Standardization..

Method engineering

Ett sätt att använda sig av flera olika metodologier, dela upp dessa i bitar och återanvända dem.

Movex

Affärssystem som Intenia saluför.

Offshoreutveckling

Utvecklingen tas ut ur landet och läggs, av olika anledningar, utomlands, t.ex i Indien.

OMG

Object Management Group, ratificerar UML.

Out-of-the-box

Något som kan och också kanske skall användas precis som det är vi leverans. Jmf. "Plug & play".

PRM

Project Risk Management, riskelimineringmetodologi.

R&D

"Research and development" (forskning och utveckling)-avdelning inom organisationer.
Svenskt term: FOU, forskning och utveckling.

Ratten

En metodologi för projektstyrning som används av WM-data.

RPG

Ett programmeringsspråk för IBMs stordatorer.

RUP

Rational Unified Process, Rationals egentillverkade systemutvecklingsmetodologi som desamma säljer.

SAP

Tyskt företag som tillverkar affärssystemet R3.

SDLC

System Development Life Cycle, t.ex vattenfallsmodellen.

Software crisis

Den "kris" som uppdagades på 1980-talet gällande att systemen kostade för mycket, hade undermålig kvalitet och tog för mycket resurser i anspråk.

Sumo

Wm-datas egenutvecklade systemutvecklingsmetodologi.

UML

Unified Modelling Language. En standard framtagen av OMG för dokumentering.

Use-case

Användarfallsdokumentation

Visio

Ritprogram utvecklat av Microsoft.

XMI

XML Metadata Interchange. En standard som bygger på XML för att överföra information om objekt mellan olika ritverktyg enligt UML.

XML

Extensible Markup Language. En standard för att märka upp data, skapad för att enklare transferera information i textform.

Referenser

- Avison D. E, Fitzgerald G(1995), *Information Systems Development: Methodologies, Techniques and Tools, Second ed.* United Kingdom: McGraw-Hill International.
- Avison D. E.(2003), Where now for development methodologies?, *Communications of the ACM, Volume 46 , Issue 1*, s. 78-82.
- Checkland P.(1981) *Systems thinking, systems practice.* Great Britain: John Wiley & Sons Ltd.
- DeGrace P, Stahl L(1990) *Wicked problems, righteous solutions: a catalog of modern software engineering paradigms.* Englewood cliffs, New Jersey: Yourdon Press, Prentice Hall.
- Easterby-Smith M, Thorpe R, Lowe A.(2002). *Management Research – An introduction. Second edition*, Great Britain: The Cromwell Press Ltd.
- Fitzgerald B.(1994), The Systems Development Dilemma: Whether to Adopt Formalised Systems Development Methodologies or Not?, *In W. Baets (Ed.), Proceedings of the Second European Conference on Information Systems, Nijenrode University, Holland: Nijenrode University Press*, s. 691-706.
- Fitzgerald B.(1996), Formalised systems development Methodologies: A Critical Perspective, *Information Systems Journal Volume 6, Issue*, s.3-23.
- Fitzgerald B.(1998), An empirically-grounded framework for the information systems development process. *Proceedings of the international conference on Information systems*, s.103-114.
- Fitzgerald B.(2003), Software development method tailoring at Motorola. *Communications of the ACM, Volume 46 Issue 4*, s.64-70.
- Funk P. J.(2000), *Reuse, adaptation and validation of system development processes*, Mälardalen: Mälardalen Real-Time Research Centre, Mälardalen University
- GDMAG, Game Developer Magazine(1998-2000), nr 1-12, 1998,1999,2000.
- Gutierrez O(1989), Prototyping techniques for different problem contexts, *Proceedings of the SIGCHI conference on Human factors in computing systems: Wings for the mind*, s. 259-264
- Hatch M J.(2002), *Organisationsteori – Moderna, symboliska och postmoderna perspektiv.* Lund: Studentlitteratur.
- Henderson-Sellers B.(2003), Method engineering for OO-systems development. *Communications of the ACM, Volume 46 , Issue 10*, s. 73-78
- Highsmith J.(2000), Retiring lifecycle dinosaurs - using adaptive development to meet the challenges of a high-speed, high-change environment, *Software Testing & Quality, July/August 2000.*

Mintzberg, H.(1983). *Structure in fives – Designing effective organizations*.
New Jersey: Prentice Hall international inc.

Ralyté J., Rolland C.(2001), An approach for method reengineering, *Proceedings of the 20th International Conference on Conceptual Modeling: Conceptual Modeling*, s. 471-484

Thompson M, Wishbow N(1992), Prototyping: tools and techniques - improving software and documentation quality through rapid prototyping, *Proceedings of the 10th annual international conference on Systems documentation*, s. 191-199

Zolnowski, J. and Ting, P.(1982) An insider's survey of software development, *Proc. 6th Int'l Conf. Software Engineering*, Tokyo, Japan, s. 178-187.

WWW-referenser

Accenture[WWW-dokument], <http://www.accenture.com/>
[accesstid:2004-01-20 till 2004-05-30]

ACM - Association for Computing Machinery[WWW-databas], <http://portal.acm.org>
[accesstid:2004-01-20 till 2004-05-07]

ADB-kontoret[WWW-dokument], <http://www.adb-kontoret.goteborg.se/>
[accesstid:2004-01-20 till 2004-05-30]

DSDM[WWW-dokument], <http://www.dsdm.com/>
[accesstid:2004-05-06]

Ekonomiska biblioteket[WWW-databas], <http://www.ub.gu.se/Ge>
[accesstid:2004-01-20 till 2004-05-07]

Highsmith J.(2002), What is agile software development?, *Crosstalk, The journal of defense software Engineering*[WWW-dokument], *October issue 2002*
<http://www.stsc.hill.af.mil/crosstalk/2002/10/highsmith.html>[accesstid: 2004-04-28]

Holtem R, Dreiling A, Becker J(2001), *Ontology Driven Engineering for information systems development*[WWW-dokument],
[http://www.wi.uni-muenster.de/is/projekte/mw-kid/pdfs/Ontology-Driven Method Engineering for Information Systems Development.pdf](http://www.wi.uni-muenster.de/is/projekte/mw-kid/pdfs/Ontology-Driven%20Method%20Engineering%20for%20Information%20Systems%20Development.pdf)
[accesstid:2004-01-20 till 2004-05-19]

IFS[WWW-dokument], <http://www.ifsworld.com/>
[accesstid:2004-01-20 till 2004-05-30]

Intenia[WWW-dokument], <http://www.intenia.com/>
[accesstid:2004-01-20 till 2004-05-30]

Keith E. R.(2002), Agile software development processes: a different approach to software design, *Agile Alliance*[WWW-dokument]. URL www.agilealliance.com/articles/articles/ADifferentApproach.pdf[accesstid: 2004-04-28]

Nationalencyklopedin[WWW-databas], <http://www.ne.se>

[accesstid:2004-05-06]

OMG – Object Management Group[WWW-dokument], <http://www.omg.com/>
[accesstid:2004-05-06]

WM-data[WWW-dokument], <http://www.wmdata.se/>
[accesstid:2004-01-20 till 2004-05-30]

Bilagor

Bilaga 1, Frågebatteri

De meningar som står parantetiserade är menade som hjälp för intervjuarna att styra samtalet i rätt riktning och lästes således inte upp för den intervjuade.

Introduktion av oss:

Vilka är vi?

Varför gör vi en intervju:

För att ta reda på om metodologier används.

På vilket sätt används metodologier.

Anpassas dessa metodologier, isåfall varför och hur?

Företags- och personfrågor:

Om du skulle beskriva vad företaget är och gör hur skulle du då beskriva detta?

Vilka är dina arbetsuppgifter inom företaget?

Systemutvecklingsmetodologier:

Vad är en metodologi för dig/ert företag(och hur många finns det på företaget)?

Vad är en god metodologi?

Använder ni systemutvecklingsmetodologier som enda verktyg? Eller finns det andra mer processövergripande metodologie ovanpå?(Visa vår modell)

Hur använder ni metodologier(följa slaviskt eller fritt)?

Anpassning av systemutvecklingsmetodologie:

Finns det anledning att anpassa metodologier? (på samtliga nivåer)

Vad, tror du, driver anpassningen? (omvärldsfaktorer, intern organisation, individens kunskap, kundkrav, uppdragets art, blivit påtvingad)

Hur anpassas er/era metodolog/metodologier? Varför?(methodengineer)

Passar vissa metodologier bättre för anpassning? Varför? (DSDM)

Bilaga 2, Intervjumaterial IFS

Transkribering, IFS, 2004-03-15

[...]=efterhandskommentarer av intervjuarna

M=Mattias Grytting

B=Björn Olsson

J=IFS

M: Ibland så är det ju lite svårt att karaktärisera företag, vi kan ju sitta hemma och fundera: dom här företagen gör ju det här och ser på sig själva på det här sättet, men det behöver inte vara sant. Pratar man med företaget så kan dom ha en helt annan bild av sig själva. Hur ser ni på er själva? Hur skulle du beskriva IFS? Vad är det ni gör jämfört med andra konsultbolag?

J: Vi är ju inget konsultbolag utan vi är ju ett programvarutvecklingsföretag. Vi bygger en produkt och säljer den. Det är en stor skillnad mot hur vi differentierar oss mot konsulterna. Sen har vi ju konsulter som arbetar med att installera vår produkt, men IFS är ju...öööhhh...vi konsulter inte på vad som helst utan vi konsulter på vår egen produkt. Vi bygger och äger en produkt, det är den största skillnaden. Vi är det programvaruföretag.. öhhh... Vi jobbar med affärssystem. Vi säljer och levererar basfunktionalitet för både industri och finansföretag kan man säga, framför allt industriföretag. Det är därför vi heter industrial...öhh...ja, industrial and financial systems. Ehhh...och vi strävar efter att bli marknadsledande i olika affärssegment och det är dom segmenten som jag visade förut[se företagsfakta om IFS]. Vi har gått från att vara ett ganska litet företag...vi är grundat i Sverige 1983 i Linköping och vi har gått från att vara ett ganska litet företag till att på dom senare åren, strax före 2000, explodera och bli väldigt, väldigt stora. Så vi är verkligen ett multinationellt företag och vi strävar efter att bli marknadsledande i dom segmenten vi pratade om förut. Var detta ungefär vad ni hade tänkt er?

M,B: Det var bra, det är inte alltid att vår bild stämmer överens med din bild med tanke på att vår bild är den vi får från tidningarna.

M: Mmmm...skall vi gå in på själva maten här då vad gäller dom delarna så....funderar vi lite på...vi har ju en klar bild av vad en metod är för oss, men det är inte alltid att man har samma definition av vad som är en metod.

J: Näää.... precis.

M: Så den första frågan lyder: vad är en metod för dig eller ert företag, vad innebär det? Vad använder man den till, vad gör man med den?

J: Det är en bra fråga. En sådan där grej skulle man egentligen ha förberett ett svar på då, som tv-kockarna... men jag får väl tänka lite granna...

B: Å, nej...

J: För mig är en metod eller...hmmm...jag får väl egentligen bara prata fritt här då.

B: Ja, det är ju det som är meningen.

J: För mig är en metod väldigt mycket ehmm...det är en hjälp för att ta sig igenom ett antal olika faser och det handlar väldigt mycket om att försöka förbereda så mycket som möjligt och att man skall få...hmm...så långt som möjligt kunna förutsäga vad det är som skall göras och i dom faserna tydligt delat upp ansvar och leverabler så att man vet vad som skall komma ut ur varje fas. Syftet med alltihopa är att man skall få hög leveransprecision, att man skall leverera det som kravställaren hade tänkt sig med rätt kvalitet, och då är metoden ett medel för att nå upp till rätt leveransprecision. Det är det viktigaste för mig. Ehh...sen handlar det också om att ha kontroll över projektet att veta vad...att veta... väldigt viktigt är att ta reda på vad man skall göra. Det är otroligt viktigt, så att man vet vad det är man skall leverera eh...på så sätt kunna lägga upp en vettig leveransplan för när man skall leverera olika delmoment och när dom skall valideras och hur man involverar kravställaren i cykeln och så vidare. Så att man har kontroll över flödet, både så att man själv har kontroll på vad man gör så att man kan effektivt fördela sina resurser. Att man kan planera effektivt så att man gör rätt saker i rätt ordning det är otroligt viktigt att...det blir annars oftast väldigt stora väntetider...ehmm...men även att man kan förbereda kravställarna på när man behöver deras input och när man behöver deras hjälp att validera lösningarna så att produkten är komplett och validerad när man släpper den. Ehh...sen en annan nyckelbit som du berörde när du inledde, det är att jag tycker att en metod är normativ, det är otroligt viktigt... den beskriver normalfallet för hur...när jag kommer inför den här situationen då skall jag inte stå helt handfallen. Jag skall veta ungefär vad som händer i den här fasen. Och hur jag skall ta mig igenom dom stegen, dom aktiviteterna som finns. Men det är normativt, det beskriver alltså vad är normalfallet och det innebär att i alla dom fallen som...ehh...där verkligheten skiljer sig från modellen, så måste man tänka själv och man måste göra dom förändringarna som krävs för just den situationen som uppstår så att man får inte följa en metod slaviskt, men man får heller inte vara för slapp i sitt utnyttjande av metoden för då är det väldigt stor risk att man tappar greppet om projektet. Så att...det är viktigt att se att det inte finns någon metod som är komplett för alla typer av händelser och styr man upp alla händelser som man tror kan inträffa, blir oftast metoden oöverskådlig så att man inte kan följa den. Så det handlar om att beskriva normalfallet...att den är normativ, det är viktigt för mig.

M: mmm.

J: Var det ett svar?

M: Det var det utan tvekan. Ehmm...om man då tittar på metodfloran i allmänhet så kan man konstatera då att...ehmm...ibland så har företag av olika anledningar flera olika metoder, anta att dom är väldigt stora eller väldigt små eller att dom är divisionaliserade eller vad det nu kan vara.

J: ja...

M: Så använder man sig av flera metoder.

J: mmmm...

M: Ehmm...så...har ni en sådan situation här, så att ni har ett gäng metoder som ni använder i olika lägen eller har ni en uppsamlade metod som i sin tur äger en massa delar av metoder som ni kan använda eller hur ser strukturen ut?

J: Vår strategi.... jag har jobbat med metodfrågor inom konsultorganisationen tidigare då, och när jag började så var inte metodsidan så väldigt väl utvecklad inom IFS utan man körde ganska så atomära projekt. Det kunde skilja sig mellan hur man arbetade mellan olika projekt på samma kontor mer eller mindre, och det där är ofantligt ineffektivt för att när vi jobbar med projekt så är det i princip alltid samma faser i ett projekt när man installerar ett affärssystem. Man skall göra en projektering och sätta scopet för vad det är som skall göras, man skall samla in kraven, man skall matcha kundens processer med processerna vi skall göra i systemet. Man skall rita dom avvikelserna som finns, man skall specia dom anpassningarna som behöver göras, man skall förbereda grunddata och installationer och så vidare. Det finns en mängd olika saker man skall göra. Och det är väldigt mycket effektivare om man har ett gemensamt angreppssätt för hur man gör för att om du då byter resurser mellan projekt eller mellan kontor eller till och med mellan länder, så har man fortfarande en gemensam bas, man förstår samma terminologi. Man vet ungefär vad det är som skall göras även ifall man då gör mindre justeringar som vi pratade om förut, utifrån dom behoven som finns i projektet. Men det är...ur min synpunkt är det otroligt viktigt att för att IFS skall vara professionella speciellt när vi driver multinationella projekt med en kund som köper vårt affärssystem som skall installeras i en mängd olika länder, är det väldigt väldigt viktigt för IFS att vi uppträder professionellt och möter kunden på ungefär samma sätt i alla dom olika länderna. Vi strävar ju efter att vi skall försöka använda dom standardmetoderna som vi har tagit fram centralt då, sen är ju det en mognadsfråga och det är en utbildningsfråga och man får ta det här i steg, men i princip så har vi en metod för konsultprojekten och vi har en metod för vår utveckling av standardprodukten och dom är ganska lika varandra för att vi just skall kunna dra fördelar utav resursenbyten mellan bolagen och sånt där.

M: För att man skall känna igen sig...

J: Man skall känna igen sig och man skall förstå ungefär hur det fungerar, men sen är det ju rätt stor skillnad på att göra ett kundprojekt och att utveckla en standardprodukt så att...därför så skiljer sig metoderna till viss del. Men dom momenten som handlar om att göra själva systemutvecklingen, det är i princip samma tänk och det jobbade jag med innan jag jobbade på R&D så jag har ju mer eller mindre tagit med mig det tänket in till R&D och byggt en...skapat en ny version av den processen för utvecklingsbolaget.

M: Som är anpassad för R&D så att säga?

J: Precis! Så att man kan säga att vår utvecklingsmetodik på utvecklingsbolaget är en anpassning av metodiken för konsultprojekten. Så redan där har man gjort en typ utav anpassning.

Sen så har ju vi R&D...som sagt, det finns ju två sidor av det myntet. Det ena är den rena forskningssidan och det andra är den rena utvecklingssidan och det jag ansvarar för är utvecklingssidan och det är ju ganska likt ett kundprojekt. Det handlar om att ta in krav, det handlar om att specia upp vad det är man skall göra, ta fram en vettig plan för det. Utveckla det, testa, validera, löpande köra systemtester, paketera och releasa produkten. Och det är ganska så...det måste man ha koll på, det gäller att ha tungan rätt i mun. Där är man ganska uppstyrd, medans ett researchprojekt är ju väldigt fritt och där vet jag att där använder man...där kan man testa lite olika metoder just för att man är i ett forskningsskede, och då är det liksom inte...det går ju inte att styra upp, eftersom man inte vet vad det är man skall komma fram till. Dom bitarna är mycket mycket friare än ett utvecklingsprojekt.

M: Men kan det vara så att du ser det som kanske till och med en fördel att vara mindre fri i ett utvecklingsprojekt för att processen i ett utvecklingsprojekt är just sådan.

J: Där är man mycket mer fri.

M: Precis det är inte så att det känns instängt heller. Utan det är ...

J: Näe precis. Nu när vi har kört några projekt här med den metoden som vi har här idag så...så känner sig folk tryggare det gör ju... dom vet vad som skall hända... Har vi en schysst plan, så vet folk att: "nu har jag gjort det jag skall och på fredag gå hem och vara nöjd". Tidigare när man inte hade någon väldigt bra strukturerad plan, då kände man alltid stress på sig att man hela tiden...hela tiden skulle lösningen växa... "har jag hunnit med allting, har jag gjort allt jag skall". Och "hinner jag testa detta nu?" och så vidare... allting blir väldigt otydligt om man inte har en schysst struktur att utgå ifrån, som folk kan känna sig trygga i och så.

M,B: mmmm.

B: Så att man vet vart man befinner sig.

J: Precis!

Och det är ju inte minst viktigt för företagsledningen och för oss att veta att: "hinner vi det vi skall? Hur ligger vi till?" Och mycket av det är ju inte alltid att man skall uppnå precis 100%, men vet man hur man ligger till, då kan man prioritera. Och man kan välja bort saker. Man kan prioritera vad det är som är viktigast att leverera och man kan fördela om resurser och så vidare. Man kan ha schysst kontroll på produkten För våra projekt sträcker sig ofta....det är inte ovanligt att dom ligger en bra bit över ett år och det drar väldigt, väldigt mycket resurser.

M: Det är ganska långt det...

J: Det är ett stort projekt.

M: Ohh ja...

J: Men vi släpper ju en gigantisk produkt, och vi pratar ju om flera hundra utvecklare, så att för att vi skall få det att gå ihop och det skall gå ihop med vår marknadsaktivitet när vi släpper produkten, så måste vi ha tungan rätt i mun när vi arbetar och det är därför vi använder den metod vi har.

M: Det är ju klart intressant då...öhh..ehhmm... Vi kanske skall diskutera det om det är ok då, den metoden, var kommer den ifrån? Från början?

J: Det är ju som vanligt då, det är ju kanske lite svårt att spåra exakta källan då. Men vi hade...hmmm...när jag började på IFS så fanns det inte speciellt hårt uppstyrt hur vi skulle arbeta metodmässigt, men det fanns en del embryon från folk som hade kommit från andra företag och så. Vi hade en del estimeringsmodeller och så som fanns i företaget sen gammalt. Och....den biten som har med själva utvecklingen att göra, om vi nu isolerar det för en leveransmetodik, den består av så mycket mer. Det handlar om olika försäljningsfaser och hur installerar du på kundparken och så där. Men om vi tittat på själva utvecklingssidan när

vi...designar och utvecklar kod... så..... så.... har vi.... i olika mognadsfaser i projektets gång, i företagets gång, kan man säga, har vi varit tvugna att ta fram verktyg för att leverera och stötta projekt. Bland annat så jobbade jag i ett stort projekt som levererade en distributionslösning till en läkemedelsdistributör. Där vi hade väldigt tuffa tidskrav på oss, och där blev vi tvugna att designa upp en utvecklingscykel, en mindre utvecklingscykel som vi kunde återanvända och lägga ut i tiden. Och utifrån det projektet så har jag sen anpassat den cykeln och putsat på den tills vi har fått en ganska komplett utvecklingsmetodologi. Vi tog även fram utvecklingsrutiner och standarder för...vi har haft väldigt många bra baser att stå på, vi har alltid använt...öhhh...configuration management systems. Så vi versionshanterar vår kod. Metoden handlade väldigt mycket om att knyta ihop alla dom här bitarna så vi vet: vad utvecklar vi? vad har vi installerat i våra egna miljöer och i kundens miljöer? och så vidare, så att man knyter ihop allt från lägsta nivå, från kodfragment i en fil till upp till in i configuration management systemet, till en spec upp i en projektplan så att man får allting att hänga ihop, så att man har spårbarhet på sina krav. Och.....dom erfarenheterna som vi har lärt oss i dom större utvecklingsprojekten, och det är riktigt stora utvecklingsprojekt på kundsidan, har vi sedan gått vidare med och förfinat, återanvänt och testat i olika situationer och sen byggt ihop till en komplett metodik. Vi har gått steg för steg kan man säga. Det har inte varit någon riktig big bang.

B: Det har vuxit fram evolutionsmässigt helt enkelt...

J: Det har vuxit fram ja! Precis.

J: Sen när vi tog över paketmetodiken som vi kallar den för...så när vi tog fram den till R&D så fanns det mer eller mindre redan ett färdigt koncept som skulle anpassas... Anpassas och det blev lite mera en större förändring då. Men....att....jag kan tänka mig att estimeringsmodeller och sådant där kan man nog återfinna på andra företag som liknar dom vi har. Och cyklerna också. Jag har hört från andra som jobbar att det finns vissa likheter med paketmetodiken och RUP till exempel. Men jag har inte när jag har tagit fram paketmetodiken, jag har inte suttit med någon annan metod framför mig och funderat på hur jag skall tweaka den, utan det har mer varit att vi har dom här, vi måste lösa dom här problemen i våra projekt och sen så har vi då, utifrån dom sätten vi arbetar, anpassat fram en metod som passar för våra behov kan man säga. Så den är organiskt framvuxen kan man säga.

M: Det innebär ju också isåfall att...man tycker att man generellt är...vad skall man säga...man tycker att det känns som det är ens egen metod på nåt sätt, att man inte har...att man har större anledning att följa den eftersom den är organiskt framvuxen, att den känns som att man känner igen sig i den på nåt sätt.

J: Ja, fast det beror ju också på att eftersom vi är en stor organisation, det är väldigt många personer som jobbar inom IFS så att du slår ju inte bara på ljuset så här [knäpp med fingrarna], och sen så tycker alla att såhär har vi alltid arbetat, men det...det är alltid en fråga om utrullning och commitment från ledningen och så där, för att en sån här sak skall fungera, då. Ehh...men jag tror ju att det är lättare för folk att acceptera en metod som är framtagen för våra behov än nånting annat som man skall följa bara för att det skall vara så, typ. Ehh...nu är det ju ingen metod som är framtagen bara för att det skall vara så, självklart, man har alltid en tanke bakom stegen. Men jag tror att det är lättare att motivera en metod som är framtagen just för våra behov. Det tror jag.

M: Sen kan man också kanske tänka sig så här att eeh...om man kontrasterar fördelarna med en sån här organiskt framväxt metod med sin bas i verksamheten kontra...kalla det för paketerad lösning, som säger att: "nu skall vi köra RUP, killar" till exempel, så kan man tänka sig att...ehh...om man nyanställer folk så kan det vara en fördel till exempel om du har RUP, så om dom har tidigare kännedom om RUP så smälter dom.....

JIn mycket lättare, ja.

M:...Än om man har en egenutvecklade metod. Hur ser ni på...är det ett problem för er?

J: Nej det är det inte. Av den enkla anledningen att vi anställer inte folk [skratt, alla]

M: Jaa...nej det är klart, då har man skalat bort den delen.
[skratt, alla]

J: Om ni har lagt märke till hur det är i IT-branschen dom senaste åren, såå, alltså, nä, det är just nu inget problem, men jag kan tänka mig att i en tillväxtfas då, så kan det säkert bli ett problem. Men, jag tror inte att det är ett så stort problem som att...det största problemet är om du har kunskap som bara sitter i väggarna och inte är dokumenterad...det tror jag är största problemet, för att jag tror att relativt fort kommer man in i våran metod eftersom att vi har ett utbildningskoncept...och ehhh...folk runtomkring dig jobbar i...på det sättet. Efter bara någon cykel så känner du igen dig, ganska fort. Jag tror att det viktigaste är att du har en dokumenterad process, att du har en metod som du följer och att du ger folk en introduktion när dom kommer då. Jag tror att rätt många av momenten i olika metoder är rätt så, ganska, lika. Terminologi och såna saker, men det skiljer sig alltid på olika företag. Jag tror inte att det är några större problem...kan du RUP till exempel, när du kommer hit går det nog ganska fort för dig att komma tillrätta i alla fall, tror jag.

M: Ehhh...när vi ändå talar om, eller ja, den övergripande metoden satt vi och diskuterade lite grann på hemmaplan, vi satt och ritade litegrann här innan vi skulle ut och intervjua så satt vi och funderade på de olika metodernas garnularitet och så...

J: Ja...

M: Är det så att det kan finnas flera, alltså en metod kan fungera på ett plan och en annan metod kan fungera på ett annat plan?

J: Ja...

M: Då satt vi och funderade på i tre plans nivåer kan man säga då, när vi talar om ehh..processövergripande metod...[plockar fram förklaringsmodell]

J: Mmmm.

M: När vi talar om metodologi och inom en metodologi kan det finnas ett antal metoder som i sin tur består av aktiviteter, alltså iterationer och så där då va.

J: Ja..

M: Ehhh...så om man tänker sig, så som jag har tolkat dig när du pratar om eran metod, är det snarare en processövergripande metod, alltså den är egentligen en projektstyrning- och samtidigt en systemutvecklingsmodell, samtidigt som den kanske till och med speciar enskilda aktiviteter i sig, den går över egentligen alla tre planen...

J: Den går som ett tvärsnitt över dom tre planen. Eehh...vi har alltså, om man generaliserar vilka lager vi har i vår metod, så har vi på lägsta nivån referensmanualerna för våra utvecklingsverktyg, dom räknas ju inte egentligen in i metoden utan det är ju: "såhär fungerar verktygen", det är ju inget som fgdgvi fn[otydligt] metodmässigt, däremot har vi ett mellanlager mellan projektmetodiken och referensmanualerna...eehh...med standards och guidelines för hur applicerar vi verktygena i projektet på aktivitetsnivå? Alltså: vad gör jag i det här steget? Inte på detaljnivå, men det styr upp sådana saker som vad har vi för namnkonventioner på...eeeh...våra paket, som det heter i Harvest, som är vårt versionshanteringssystem. Alltså: hur namnsätter jag det och hur gör jag i den här situationen? När höjer jag status till en ny status i versionshanteringssystemet? När gör jag det, jo, när jag levererar från vår utvecklingsmiljö till vårt interna hgvfgffdes[otydligt], då höjer vi status och den biten, det är ett mellanlager som förklarar hur vi använder tekniken, och det är ganska hands on på lägsta nivå. Vi har alltså standarder och instruktioner på den nivån, hur vi använder systemena, men det är ingen referensmanual. Ehh...sen nästa steg är ju på aktivitetsnivå, vad som händer i olika processer, och sen på toppnivå så visar vi då hur hela R & D:s huvudprocess fungerar...så att...jag vet inte...vi kanske kan kika lite...bara så ni får en känsla av hur det ser ut på...vart skall jag börja....? [visar Powerpoint-presentation om processen] Det här är bara för R&D, sen har vi en process för hela företaget som helhet också, men det här är för R&D. Och där har vi då en request management-process som egentligen är kravinsamlingsprocessen, sen har vi utvecklingsprocessen, releaseprocessen, det är dom här två som jag är ansvarig för. Sen har vi en supportprocess som handlar om att ta emot fel i vår produkt utifrån länderna, alltså fel i standardprodukterna som skall servas tillbaka med patchar. Och sen så har vi översättningshantering som löper parallellt här och vi har release, marketing och communication då, som kommunicerar budskapet kring våra nya produkter då, så att det här är då vår toppnivå. Tittar man på utvecklingsprocessen som sådan då, så är den ju på högsta nivå då fasindelad i dom olika faserna vi har för projektet med en projektplanering där vi sätter samman projektet. Som input där kommer ju kraven från kravprocessen. Vi har nånting som vi kallar för implementation study, projektering brukar vi säga på svenska, en designfas där vi...eehh...bryter ner kraven i sina beståndsdelar och estimerar dom, funkspecar dom och godkänner dom med kund, och sen gör vi en plan, en paketplan som vi kallar den för, som vi sen utvecklar då, paket för paket. Paket är alltså...eehh...vi itererar flera gånger under projektet, ett antal olika leveranser...öhh...och till slut så börjar vi testplanera och sen kör vi systemtest innan vi går in i releasefas. Sen så har vi ett antal olika stödprocesser här för projektledning och ett antal olika riskhanteringspunkter då som inträffar i olika ställen i processen... Sen så om man gräver sig ner här på lägre nivå så finns det ju relativt detaljerade instruktioner. Vi kan ju ta ett exempel här. Nu är det en ganska grund process det här, men det är kanske ett bra exempel. Här visar vi egentligen bara enkelt att vi har ett antal olika krav. Som kallas för "id" och då är det delprojektledaren som skall assigna olika designers och sen skall man då lägga upp den här i våra projektsystem och i våra buggrapporteringssystem och vi skall även ha scopelista som vi kommunicerar mot våra kravställare och tittar man på en sådan här punkt då. Så finns det instruktioner, dels i text i vissa fall, men ganska ofta då när det är fråga om detaljer så finns en direktlänk till vår methodhemsida som förklarar hur du gör det här mer detaljerat. Så att processen och instruktionerna hänger ihop.

M: Det är egentligen ner till screenshots i verktygsnivå...

J: Precis! Mer eller mindre.

M: Tjosan!

J: Så att för att man skall lägga upp då, det här fallet handlar om hur lägger jag upp en utvecklingspunkt i våra projektsystem så att vi kan följa upp på den. Vi distribuerar väldigt mycket av våra uppgifter till dom enskilda medlemmarna då. Och då får dom instruktioner kring hur dom skall göra här, så dom kan antingen då gå in på vår metodhemsida där vi då har olika standarder, t.ex den här som vi pratade om nu, ”skapa en id” eller att man navigerar från processen för att ta reda på vad man skall göra och får instruktioner den vägen då. Så att...sen så ser ni här också att vi har roller kopplade till alla aktiviteter... Ett sådant här streck då[pekar] innebär att man är flera stycken om det då. För alla aktiviteter finns det ett definierat rollansvar. För att man skall vara säker på att saker och ting blir gjorda och inte faller mellan stolarna och att man själv skall kunna känna att om jag har...man kan ju ha flera roller självklart...men att jag vet att det är dom här bitarna jag skall ta ansvar för. Och dom andra sakerna kommer någon annan att fixa åt mig. Det är också väldigt viktigt om man skall kunna sätta upp en bra projektorganisation att man vet att man har dom kompetenserna man behöver och att man kan matcha tjänster med dom rollerna som krävs i projektet. Så att...

B: Det har tagit ett litet tag att ta fram den här metoden. Det är inte gjort över en natt direkt...

J: Det är inte gjort över en natt, kan man inte påstå...ehmmm...och vissa av dom här processerna är ju mycket djupare också. Men vi har ganska bra koll på vad vi gör nu som den här modellen används ju framför allt som referensmanual. Det är inte så ofta någon i projekten använder den här dagligen. Men det är fortfarande en väldigt bra struktur, för vi ser ju var vi är och vilka ansvar vi har, vilka saker vi skall utföra. Det är näst intill omöjligt att få bra koll på det om du inte ritar upp det. Så att det är så vi har gjort...

M: okej...det blir klart intressant, för nu kommer vi in på sådana här saker som alltså...det här är ingenting som man använder dagligen och då menar du egentligen att man inte arbetar efter den dagligen men man har en klar uppfattning vart man är på väg inom den stora ramen egentligen?

J: Alltså, man följer dom här stegen men man behöver oftast inte... alltså vi har ju utbildningar på vad man gör i dom olika rollerna o.s.v så att man behöver kanske inte gå in och titta i modellen för att...om vi tittar på hur t.ex utvecklingsprocessen ser ut då... För att t.ex skriva ny kod så finns det ett antal olika grundbitar som skall göras. Det här vet ju en utvecklare att han skall göra, han behöver ju inte gå in och titta här. Däremot så om man är osäker på vad som ingår i olika steg så kan man gå in och kika här, liksom.

M: Men det är egentligen...vad skall man säga...det har blivit rutiniserat...en utvecklare vet hur man skriver kod, han behöver inte kolla i schemat vad han skall göra utan han har klart för sig det. Han vet vad slutpunkten skall vara.

J: Precis!

Däremot så vet vi ju...sen så styr vi ju...det här är ju bara processen då sen så har vi ju...vi styr ju upp det här projektmässigt också i våra projektsystem också, för varje punkt vi lägger upp så har vi en tidsplan och ett antal uppgifter som skall göras. Dom bockar vi ju av och mäter

löpande under projektets gång så att vi ser ju vad som blivit utfört och inte, men processen är ju väldigt viktig för att vi skall veta vad det är som ingår. Och att vi själva kan ge folk rätt utbildning och rätt instruktioner så dom vet vad dom skall göra. Så det här är ju mycket ett verktyg för mig och dom som jobbar med processen. Men enskilda projekmedlemmar är nog sällan inne och tittar här tror jag. Däremot delprojektledare och så som är ansvariga för olika faser, dom har nog hjälp av att titta här ibland. Och det är bra också om man skall gå igenom en ny fas och aldrig har gjort det förut så kan man gå igenom aktiviteterna och man kan få bra instruktioner när man behöver dom genom att gå in i metodwebben då. Så att ehmm...frågan är hur djupt vi skall gå in, för jag har ju väldigt bra presentationer som går igenom hur åra cykler fungerar och så också.

B: det är väl egentligen inte så intressant för just vårt arbete.

J: Nä.

M: Det här tycker jag var klart intressant, men det innebär egentligen att den metoden som ni använder fungerar egentligen på alltifrån hur ni sätter samman ett projekt till kodningsstandard mer eller mindre?

J: Japp. Kodningsstandarder har vi ju alltså...men det förklarar inte hur man skriver koden, utan hur man associerar koden, hur taggar jag koden o.s.v. Hur hanterar jag versionshanteringssystemet? Referensmanualerna för hur verktygen fungerar i sig, det är liksom rena referensmanualer som vi inte lägger oss i. Men vi har alltså instruktioner på lägsta nivå för hur man hanterar en kodcykel, för att alla leveranser skall fungera. Vi har ju väldigt många utvecklingsmiljöer som skall byggas ihop och dom skall leveras till olika testcykler och då måste man ha en schysst struktur som beskriver detta så att alla gör på samma sätt så att det överhuvudtaget skall gå att testa produkten under projektets gång. Då måste man ha koll på detta. Så vi styr upp det från den nivån upp till vilka faser vi har och hur man tar sig igenom det då, så det är både hur man sätter ihop projektet men också faser innan o.s.v. Hur vi samlar in kraven och hur vi initierar projektet. Hur vi sätter samman organisationen, bygger upp vår infrastruktur för att kunna köra projektet. Hur vi designar kraven...estimerar dom, bryter ner och estimerar dom och hur man gör en plan för att kunna återupprepa dom här cyklerna...det här är en utvecklingscykel här då[pekar]. Sen återupprepas den här, vi kör ungefär fyra veckors utveckling i taget, sen testar vi och kundvaliderar och sen så börjar nästa cykel direkt där på och sen så hanterar vi rättningen parallellt med nästa utvecklingscykel. Så vi har en överlappande modell som blir ganska effektiv då.

B: Okej...öhh...hur fungerar...åå...det var intressant...om ni kör fyra veckors utvecklingscykler, hur fungerar då, det är klart att man måste anpassa dom enskilda bitarna som inte ingår i den här cykeln så det finns en möjlighet att dom blir körbara efter fyra veckor. Hur gör ni den uppdelningen där?

J: Precis...ehmm det är ju...allt det här utgår ifrån att du kan förutsäga ungefär vad du skall göra. Att du har en estimatmodell som bygger på att vi funktionellt specar upp vad det är vi skall göra och det är väldigt mycket på vad-nivå. Vad är det som skall göras? För oavsett hur väl vi specar så kommer alltid detaljerna att börja röra sig när vi börjar jobba med lösningen. Om vi har tagit fram en relativt grov spec, men den talar om vad det är du skall göra, vilka moment. Sen bryter vi ner det. I en estimatmodell så bryter vi ner specen i viktiga funktioner så att vi kan se vilka delfunktioner som ingår. Sen använder vi oss av komplexiteter som vi sätter på dom olika nivåerna, är det en för hög komplexitet så måste vi bryta ner det i finare

funktioner och gå tillbaka till kravställaren för att få mer information. Och komplexiteten styr också då ungefär hur mycket programmeringstid vi tror att det är. Sen arbetar vi med påläggsfaktorer som är...talar om vad vi måste göra förutom att programmera. Man skall skriva testprotokoll, man skall testa och man skall göra en hel massa saker. Man skall versionshantera koden, man skall göra massa saker. Vi har påslag för att få en bra bild, och i slutändan så får vi en ganska stor lista med krav och nedbrutna strukturer med tidsestimat. Och där har vi då i vår planeringsprocess, en analysprocess där vi försöker para ihop krav som hör ihop...och där man försöker att dela in dem i områden tilldela dem till olika team. Och...syftet med det då är att man skall ha liknande funktionalitet i samma team så att man kan fortsätta med...att man har en bra kompetensmatchning, att rätt team gör rätt sak, att man fortsätter med samma saker...och...och då måste man börja fundera på i vilken ordning man gör de här sakerna, för att det finns ju liksom kronologiska beroenden emellan olika krav, att jag måste först göra det här för att kunna göra det andra och så vidare. Och det man gör då är...i vissa fall, eftersom cykeln är...den är konstant. Det innebär att om det är smågrejor skall man kunna få in flera saker i en utvecklingscykel. Är det stora saker så kanske man måste bestämma sig då för att jag gör basdatan i första paketet och sen så i andra paketet så fortsätter jag med logiken. Det viktigaste är liksom att man vet vilka delfunktioner det är. Då kan man då bedöma vilka delfunktioner som går att leverera. Sen kanske man inte kan testa allt, men man kan testa så långt man har kommit och det är jätteviktigt. För att om man inte gör den uppdelningen utan bara...man måste göra klart allt innan man testar...om man har den inställningen så kan man egentligen inte leverera nånting förrän projektet är färdigt, och då ligger man jävligt risigt till om man bygger en produkt med 300 utvecklare som håller på ett år, liksom.

M: Och så funkar det inte när man kör det sista och så....

J: Nej, så att vi delar liksom in...så att man får ta eh...bita upp det så att vi vet vilka delfunktioner som ingår, ganska väl. Så kan vi bita upp det ganska fint också. Sen blir det ju så att...i de olika cyklerna då får man ibland lyfta ut funktionalitet som man inte hinner med till senare paket. Men det är fortfarande viktigt att man levererar i tid eftersom att vi vill att alla testar samtidigt annars går det liksom inte att testa. Om det är utveckling igång så går det liksom inte. Speciellt inte när vi har utveckling igång i 3 världsdelar då. Så att...eehh...

B: Jag tänkte på det att....det står "last modified" där...[pekar på skärmen]. Uppdateras dom här modellerna, och isåfall vad är det som driver den uppdateringen..och vad är det som uppdateras? Är det bara att det har varit nåt stavfel eller är det så att det har kommit in nya....rutor och bubblor om man säger så?

J: Både och...i vissa fall så märker man att..eehhh....editoriella förändringar, dom sker ju hela tiden, när man upptäcker småfel....men nu är det ju såhär att vi har ju precis nu ställt om vår marknadsprocess till att bli mer marknadsdriven. Vi har ändrat kravinsamlingsprocessen så att de här industrisegmentena som jag pratade om i början...vi har alltså marknadsavdelningar som jobbar mot de industrisegmentena, det är framförallt dom som ställer krav på vad som skall in i våran produkt. Så att...i samband med att vi ställde om utvecklingsprocessen så gjorde vi ju ganska stora förändringar i processen. Och...men det var ju i projektform och tog i princip ett år att få igenom de förändringarna. Att designa hur vi skall arbeta, och ställa om utvecklingen efter det. Ehh...och det är ju typ utav...nu har det hänt någonting i våran affärsmodell, vi måste ändra modellen, utvecklingsprocessen. Sen har vi...och det är två varianter: den editoriella och så den här "oj...våran omvärld har förändrat sig, vi måste hantera detta". Den tredje biten, det är väl egentligen när vi upptäcker att nånting inte är tillräckligt

tydligt, eller att vi behöver mer stöd eller att vi måste reda ut nånting mer i detalj. Då tar ju vi ett race och försöker diskutera igenom hur vi skall arbeta här. Man snackar med dom som arbetar i projektena och försöker strukturera upp det. Ofta är det här behovsdrivet från projektet.

M: Det kan vara så till och med att en enskild projektmedlem har kommit på nåt att...det funkar inte som jag vill att det skall funka här...har vi ett problem? ...och så tar man upp det?

J: Framförallt är det ofta såhär: "hur gör vi här då? Det här har vi inte tänkt på.." Såna här smågrejor dyker upp...och det dyker upp nya saker som...jaa...ploppar upp, av mindre karaktär. Då löser vi det. Sen försöker vi rulla ut det i projektet.

M: Kan man göra så att man permanentar en sån lösning som dyker upp...om det dyker upp en sån här situation till så skall man göra såhär, för då vet man att det är...?

J: När det dyker upp saker så försöker vi liksom att ta in dom i processen löpande...men det krävs...det är ju en förankringsprocess innan man kan trycka in det då. Men ofta så...men metoden är inte...den är hyfsat komplett. Vi vet ungefär vad vi skall göra, men den är aldrig färdig.

M: Den är levande hela vägen?

J: Den är levande. Den är aldrig färdig, så att när det dyker upp saker som ehh...vi uppgraderar våra interna system, okej hur skall vi arbeta nu, ja nu måste vi ju justera det. Då får vi ju jobba med...hur skall vi hantera denna situationen nu...då får vi ändra i våra instruktioner och i vår process i vissa fall då för att det skall passa. Ehhh...och det gör vi löpande.

B: Ni har ju ett antal olika segment som ni levererar mot. Du sa bland annat försvarsindustrin och du nämnde turism.

J: Japp. Hmmm...näe inte turism.

B: Näe...aviation var det. Ser du någon skillnad i behov av utvecklingsmetoder vad gäller t.ex defense och vad skall vi ta i andra ändan.... processindustrin?

J: Jaaa..... Nääää, inte egentligen.

B: Går det att köra samma utvecklingsmetod oavsett vilken bransch det är?

J: Ja, i princip kan vi göra det... alltså, det är fortfarande samma produkt och samma resurser som jobbar med det och ett krav är ett krav och vi skall införa det och det skall lösas. Vi har egentligen samma...vi har egentligen samma krav på oss som leverantör att leverera hög kvalitet och att vara i tid och så vidare oavsett om det är...processindustrin som skall ha det. Ett kärnkraftverk är...vi har rätt mycket av drifterna av underhållssystemen för kärnkraftverken och det är minst lika viktigt att det fungerar som när vi levererar till flygplanstillverkarna.

B: Hur är det med försvarsindustrin för dom har ju...nu antar jag igen...spekulerar men...dom har förmodligen redan fasta krav på vad det är dom vill gentemot om ni skulle göra någonting emot t.ex... hmmm...

J: Det ligger något i det du säger. Vi har faktiskt ett joint operation-företag tillsammans med BA Systems i England...som är en erkänd försvarsleverantör och det företaget heter nu mera IFS Defense även om det är ett joint operation företag då. Emmm...och dom har väl lite specifika rutiner för... hur dom hanterar sin utveckling, men i princip tar dom vår core-produkt och bygger tillägg för den kan man säga och det är ingenting som jag ansvarar för med min metod. Men jag tror att dom jobbar nog ganska likt oss ändå, det som är skillanden är ju då säkerhetsklassningen ... vem får veta vad, och så vidare då, va. Man kanske har en mera sluten informationsspridningsmodell då. Men det finns egentligen ingenting som hindrar att dom jobbar på ungefär samma sätt. Det kan vara lite svårare att få in krav i och med att det är sekretessbelagt, men så specifika är inte vi...på R&D utan vi levererar t.ex då underhållsdelen av vår produkt väldigt mycket till försvarsindustrin. T.ex så levereras vår underhållsmodul med JAS för att serva och så vidare, och det är ju underhåll som underhåll. Sedan så finns det då olika försvarsspecifika tillämpningar som tex BA Systems kan bygga ihop.

M: Mmmm...så skräddarsyr man på det sättet då.

J: Eller så gör man det i kundprojekten. Det är också ett sätt att hantera det.

M: Skall vi se...det var något annat jag tänkte på...nu skall vi se.. Jo det här med organisation kontra metod.

J: Ja...

M: Man kan tänka sig att man bygger en metod som passar organisationen eller så kan man tänka sig att man bygger en organisation som passar metoden.

J: Ja...

M: Finns det så att säga något tänk kring den delen...?

J: Ja det gör det... dom hänger ihop kan man säga... om vi bara gör så här...[knappar på laptopen och visar]. Det här är vår grundorganisation för ett projekt, och vi har även grovt ritat hur den här interagerar med vår linjeorganisation...men...den här organisationen är ju framför allt framtagen för att serva alla funktionerna vi behöver göra i projektet, det är ju hur vi når målet med organisationen och det är ju därför vi arbetat med roller och inte med tjänster. Alltså om du har en tjänst som systemutvecklare så....så...måste du...det är en ganska stor tjänst att jobba som systemutvecklare. Du kan inte ta på dig flera olika roller i ett projekt...så därför har vi en rolluppsättning så att man flexibelt skall kunna sätta till folk ur linjeorganisationen till projektorganisationen. Så vi har en linjeorganisation som äger resurserna och den är indelad enligt dom produkterna som vi levererar och det är för att t.ex för att vi håller kompetenserna uppdaterade inom dom områdena att: "jag jobbar på inte distributionssidan och här har jag mina resurser som kan detta" och så sen så skapar vi ju upp projekten och bemannar med folk ur linjeorganisationen och dom får olika projektroller. Men den här rollkartan ser lite annorlunda ut än vad den gör på t.ex consulting. Och anledningen till det är att vi har en annan situation med andra tjänster. Vi har en organisation där våra

utvecklingsteam till stor del sitter på Sri Lanka och kanske inte har exakt samma...ehh...breda kompetens som våra utvecklare i Sverige har. Kanske inte både klarar av att designa, administrera och utveckla utan där kanske man är lite mer nischad och det innebär att vi har varit tvugna att ta fram roller som är mer specifika för R&D...men i och med att det är roller så kan man ändå kombinera ihop dom på flera... på samma person utifall att man har kompetens som sträcker sig över flera rollområden. Så ja, vi har varit tvugna att anpassa vår projektorganisation till linjeorganisationens eller till vilka kompetenser som finns i linjeorganisationen. Men projektorganisationens mål är att fylla dom funktionerna som krävs för att leverera projektet...Var det svar nog?

M: Ja det får man lov att säga.

Jag funderar lite granna på...om man inte är medlem i linjeorganisation och inte har någon roll i ett projekt...vad gör man då?

J: Mmmm...då jobbar man antagligen i vår supportorganisation. Det är det absolut vanligaste och supporten drivs linjemässigt eftersom den pågår konstant...så att dom flesta som jobbar i linjeorganisationen är antingen chefer eller jobbar med support. Sen har vi några..vi har t.ex några...det finns en designtjänst som är ansvarig för att hålla ihop konsistensen i en produkt t.ex...ta t.ex vår lagermodul den används utav väldigt många andra moduler som interagerar med lagermodulen och då har vi en linjetjänst som egentligen inte får beläggas i projekten utan som har koll på den och vet hur den fungerar och kan stötta projekten och göra designreview och så på våra specar och codereview på det som är utvecklat. Även om det går flera projekt parallellt så kan den personen...han binds inte upp i projektet utan han går från linjen med syfte att granska det som utvecklats så att det blir förvaltningsbart och hänger ihop, även efter det att projektet är färdigt. Det finns ett antal linjeroller som stöttar utanför projektet, sen är det även dom som äger resurserna, vi har produktchefer, supportchefer och utvecklingschefer för dom olika områdena som sitter i linjen. Men i princip, om man skall generalisera, så sitter folk antingen och jobbar i projektorganisationen eller så jobbar dom med support. Det är våra största områden inom R&D.

M: Så det är inte så att folk går ut och in i linjen från projekt så där jätte ofta?

J: Om man säger så här... Dom personerna som är i linjen dom är i linjen och dom som är i projekt dom är antingen i projekt eller i support. Men vi har folk som går mellan support och utveckling, där har vi en viss cirkulation. Men...dom rena linjetjänsterna är i allmänhet inte involverade i projekt eller i support rent operativt.

M: Ehmmmm...jaaaaa... då skall vi se vart är vi någonstans nu då...[tittar i papperet]. Vi tänkte så här.... strategisk nivå [pekar på modellen], och här talar vi då alltså om den övergripande metodologin som t.o.m hanterar projekt. Alltså organisation av projektet. Vi talar taktisk nivå och då talar vi så att säga på enskild projektnivå: "nu gör vi så här i det här projektet" och vi talar på operativ nivå där vi säger "ok...i det här projektet skall dom här aktiviteterna ingå i dom här faserna". Så stt det blir en finare granularitet ju längre ner man kommer, så att säga men det har vi vart inne på lite granna och diskuterat.

J: Och jag tror att vi opererar på ett tvärsnitt i modellen.

M: Ni hade ju så att säga en kanske heltäckande metod som både.... från ax till limpa om man skall kalla det för det.

J: Jag skulle nog vilja säga att vi har en ax till limpa-lösning.

M: Ja, det var väl ungefär där vi var.

B: Vi kan väl bolla vår teori lite då...vad det är som driver anpassningen på dom olika nivåerna? Vi trodde att det på den strategiska nivån var sådana saker som omvärldsfaktorer, hur ser verkligheten ut, vad använder våra konkurrenter för metoder.. kan vi lära oss någonting av dom?

J: Ja, vi kan ju ta exemplet som jag drog förut, att vi bestämde oss för att fokusera på industrisegment, ja då måste vi ställa om utvecklingen så att vi...totalt fokuserar på att kravkällan i de här segmentena och hur skall vi då fördela våra utvecklingstimmar mellan dom? Och hur skall kraven komma in och hur förankrar vi det under utvecklingen? Så, för oss var egentligen ett beslut på strategisk nivå att bli MER industrifokuserade än vad vi var tidigare, det är ett typiskt ett sånt strategiskt beslut som ändrade vår process på en övergripande nivå. Så...från mitt perspektiv så är din utsaga korrekt, om man säger så.

B: Vi hade ju andra teorier där också, fast dom spelar väl inte riktigt in...att det t.ex kan bero på att man blir påtvingad ny metodologi helt enkelt, p.g.a att man blir uppköpt...

J: Så är det.

B: Då har vi ju det BA Defense Systems, eller vad det var de hette.

J: Där är det inte riktigt så tight att vi fick på oss....dom...det är ju ett eget bolag. Men vi har ju köpt upp ett antal olika företag, som kör vår metod, och det inte helt lätt att få folk i Tucson, som sitter mitt ute i öknen i Arizona, nio tidszoner bort att spela samma match som vi gör. För dom har ju varit tvugna att köpa läget lite mera, kan man säga. Så att den situationen finns också. Men så länge som vi är IFS och inte blir uppköpta av någon annan, så är det mer så att vi köper upp andra. Det är ju mera vi som är den spelaren i det här fallet, då.

M: Och sen har vi också diskuterat det här, men vi var också inne på att på den taktiska nivån finns beslut om att ändra metoden efter organisationen eller organisationen efter metoden-varianten. Och sen funderade vi också litegrann på operativ nivå...om man antar att man har en metod som specar allting utom egentligen aktiviteter och så säger man att dom som separata medlemmarna i projektet, eller dom som ingår i organisationen, som ingår i den här metoden, dom har frihet att välja här...hur dom vill implementera vissa saker..."jag vet vad det är jag skall göra, men folk skiter i hur jag gör det, så länge det blir gjort på det här sättet, så blir allting bra"...eehhh...där upplevde vi litegrann att ni hade...ni specar ändå ganska hårt, ned på detaljnivå, men det finns en liten frihet för dom som...

J: Friheten ligger i att...i och med att vi specar ganska...vi specar funktionellt vad det är som skall lösas, så lämnar vi väldigt mycket frihet till teamen att göra sin egen lösning, hur det här skall implementeras i produkten. Där vill vi att utvecklarna skall vara drivande i att hitta bra lösningar. Alltså om vi säger: "jag vill ha det här resultatet", så kanske det finns 4-5 sätt att komma fram till det resultatet. Och är det då en designer med applikationskonsultbakgrund, det är en tjänst vi har, mer marknadsnära, oftast inte utvecklare, har inte alls lärt sig utveckling i vissa fall, dom kan ju ta fram en lösning som dom tycker är optimal, men som suger när man väl börjar jobba med lösningen tekniskt. Där har vi...utvecklingsteamet har stor frihet att välja hur man skall lösa uppgiften rent lösningsmässigt. Det vi styr upp, däremot, det

är ju när dom skall checka in dom här grejorna i Harvest, vårt versionshanteringssystem, då måste dom ju använda vissa namnkonventioner, måste följa vissa av dom statusslingorna och så där. För att våra byggen skall fungera när vi bygger ihop systemet. Det är framför allt dom bitarna vi styr upp. Och det måste vi styra upp, annars fungerar ingenting. Vi kan ju inte jobba centralt, vi kan inte bygga ett stort affärssystem om folk sitter i sina egna små utvecklingsmiljöer och pillar. Då går det aldrig att få ihop produkten sen, så att det vi styr upp är mera....

M: Vad, men inte hur....

J: Vad måste ni göra, ja, och i vissa fall hur, när det gäller såna standarder som gör att infrastrukturen i projektet skall fungera, men sen bestämmer inte vi exakt hur dom skall få fram en funktion, det kan dom som skall programmera göra på det smartaste sättet. Sen så jobbar vi med review istället, så att vi ser att vi itererar och testar, vi kollar i dagliga byggesmiljöer och så vidare att den som har designat kravet går in och kollar att "det här blir skitbra, har ni tänkt på det här?" och så jobbar vi liksom iterativt under utvecklingscyklerna med att...exakt hur vi skall ta fram lösningen. Men vi är relativt strikta på vilka handhavanden som måste ske för att våran interna infrastruktur skall fungera.

B: Hur är det med dokumentation nere på den operativa nivån? Ställs det några krav på programmerare, på vad dom skall lämna ifrån sig, rent dokumentationsmässigt, hur dom har löst uppgiften?

J: Ja, det gör det.

B: Och är dom specade hårt enligt.....

J: Ja.

B: Om vi nu tar nåt....hjälp mig....

J: Jag tror jag vet vad du är ute efter och jag kan förklara...

Det vi gör är att vi jobbar framförallt med den funktionella specen....om vi....flera nivåer i hguryhr[otydligt] dokumentationen...jag bara förklarar övergripande då.... Den funktionella specen är egentligen det som styr hela processen, det är liksom vad det är vi skall göra och det är ju signat av våran kravställare, då. Och den förfinar vi när vi utvecklar och det gör ju att utvecklingsteamet, när dom väljer implementationlösning, så uppdaterar dom detaljerna för att det skall stämma överens. Och det är ju då en grund för att vi skall kunna skriva användardokumentation, att den här specen är uppdaterad med hur det verkligen blev, hur det verkligen fungerar. Sen skriver också utvecklarna fönsterbeskrivningar under utvecklingens gång. Det är en del som ingår i utvecklingsarbetet, att skriva window-description. Fönsterhjälp, för just det formuläret man utvecklat. Sen har vi en roll i projektet som är dokumentatör, som jobbar med att...när utvecklingscykeln är färdig så fortsätter dokumentationscykeln med att man hjälper till att testa för att lära sig lösningen, har den uppdaterade funkspecen som input när man skriver dokumentationen, och där lägger man på mycket mer, man uppdaterar våra processmodeller ungefär som ni såg för utvecklingsprocessen, men för produkten. Man går igenom about-beskrivningar, alltså hur använder jag den här funktionaliteten i olika sammanhang, man lägger mycket fler dimensioner kring dokumentationen. Och det görs liksom då steget efter varje cykel, så det kommer liksom cyklistiskt, att man dokumenterar varje utvecklingscykel. Sen har vi ju fler sätt

att dokumentera, vi har en implementationsspec också, som är mer teknisk spec, men det är inget dokument som lever, utan det är mera till för att man skall ta fram en grov objektmodell och så där. Att man tänker efter innan man börjar hacka. Det är inget dokument som lever, och det är ingenting som används som dokumentation i senare skeden. Det gör däremot funkspecen, den....

M: Hur tas den, kalla den för den tekniska specen, hur tas den fram?

J: Den tas fram av teamet.

M: För sin del.....?

J: För sin del, ja. Och den...det den går ut på egentligen är att man tänker efter hur man väljer att implementera det här, det gör man liksom i varje cykels början. Vi brukar i och för sig förbereda den i designfasen, så att man har förberett den, så man vet att det fungerar grovt tekniskt. Sen sätter man sig ner tillsammans och ritar upp objektena och funderar på liksom hur det här skall se ut, så att man har tänkt efter innan man börjar hacka. Börjar man bara hacka direkt...ja du vet...efter en vecka så upptäcker man att faan...inte rätt nyckelstruktur i dom här tabellerna, jag måste ta bort allting och börja om från början. Det är mycket effektivare att tänka till före. Men det viktigaste där är att man har gjort det, tänkt efter före. Sen behöver man inte uppdatera den under projektets gång, för i och med att vi versionshanterar koden, i och med att vi har....vi kan...vi kan ta fram koden för en viss anpassning, när som helst i projektet, i vilken miljö som helst...eeh...och säga att det är exakt det här vi har levererat. Då är ju alltid koden svaret på lösningen, rent tekniskt. Det viktiga att dokumentera och spara undan, det är ju den funktionella beskrivningen, vad var det vi ville uppnå, hur skulle det fungera?...så att man har det som en bas för att kunna validera sen. Så att vi använder funkspecen och testprotokoll när vi testar, vi har två testinstanser i varje cykel. Dels testar vi internt, där vi tar någon intern resurs på R&D, som inte har varit med i utvecklingen, som testar efter testprotokoll som vi tar fram parallellt med utvecklingen, bara för att vi skall kunna bocka av att vi gått igenom alla nya steg i funktionaliteten, nya, förändrade steg. Vi avlusar skitfelena innan vi släpper produkten för kundvalidering. Då är kundens syfte att validera att kraven har blivit uppfyllda.

M: Funktionalitetskraven?

J: Funktionalitetskraven. Vi försöker ju hantera den egna, interna hygien. Det är ju det vi försöker lösa själva. Det är ju mycket mer effektivt att ta hand om det själv, innan man börjar iterera med ärenden fram och tillbaka, va, mellan kravställaren och oss. Men då har ju kravställaren både funktionella specen att validera emot, det blir facit över hur vi har kommit överens om att det skulle fungera.

B: Har vi byggt rätt system helt enkelt...?

J: Sen kanske dom upptäcker att den filen inte stämmer, men då vet man i alla fall det, men det är det som är vårt kontrakt, där man säger som att "det här hade vi bestämt att så här skulle det fungera tillsammans".

M: Ni levererar efter den kan man säga...?

J: Sen har ju kunden även testprotokollet att använda för att ta sig igenom testfallen då. Som en guide. Dom återanvänder testfallen från interntestet. Så att...dokumentationen som vi lämnar ifrån oss, under en sån här cykel är fönsterbeskrivningar som utvecklingsteamet skriver, testprotokoll, en uppdaterad funktionell beskrivning och det är den on-line-dokumentationen som kommer paketerad med produkten. Till det kommer även träningsmaterial som vi skickar med produkten.

B: Anledningen till att jag frågade var, kanske inte riktigt hörde hit egentligen, men det är ju ganska lätt att dokumentera ihjäl sig.

J: Ja, det är det.

B: Du svarade väl lite på den frågan också. Utvecklingsteamet gör en skiss, på hur det är tänkt att det här skall göras...

J: Ja.

B: ...men den skissen uppdateras inte sen, utan det är så att säga....det är en utgångsskiss.

J: Japp. Vill man jobba med den så gör man det, men det är utifrån eget behov, liksom. Det är inget vi kräver skall uppdateras eller nån annan skall...man kan ju plocka fram den där och....återanvända dom för att....”hur var det vi tänkte när vi gjorde detta?” Men kan inte vara säker på att den är korrekt, men man kanske kan få en fingervisning, för vi sparar ju, men vi har absolut inget krav på att det skall vara uppdaterat. Det är samma sak med våra funktionella specar. Dom stämmer när vi släpper produkten, så vi levererar faktiskt med dom till consulting så att dom har våra funktionella beskrivningar. Det är väldigt bra för dom som skall göra anpassningar till exempel. Men sedan uppdaterar vi dom inte. Då börjar vi på ny kula i nästa core-projekt, så dom dör i och med projektets död. Men det som lever är ju själva on-line-dokumentationen till produkten, där finns ju det mesta, i den funktionella specen.

M: Precis. Och uppdateras det där, reflekteras det i on-line-dokumentationen...

J: Precis. Så att det är egentligen den som är det levande dokumentet. Men det som kan vara bra med att kunna plocka fram funkspecen är att man kan få lite mer bakgrund. Hur tänkte vi och...varför gjorde vi så och så. Det kan stå i funkspecen och det kan vara bra om man skall bygga vidare på lösningen till exempel.

M: Vi går tillbaka litegrann till det här....den här tekniska delen som teamet tänker ut, hur dom skall göra saker och ting. Har dom full frihet att välja på vilket sätt dom gör det här tänket?

J: Ja, så länge dom uppfyller behovet i den funktionella specen så är det så. Men det reviewas av den personen som har skrivit funkspecen. Det reviewas hela tiden under utvecklingens gång, vi har spec-möten mellan teamet och den som har skrivit funkspecen för att stämna av att lösningen växer fram på rätt sätt. Då kan man på ett tidigt stadium säga att...jaa...det här är nog bra men jag hade nog tänkt mig lite mer så här...eller om vi gör så så får vi det här problemet längre fram. Och teamet kan också liksom så att man...vi börjar våran utvecklingscykel med att designern för de funktionella kraven träffar teamet och har ett uppstartsmöte, nu kan det ju ske via NetMeeting och så där eftersom vi sitter i olika zoner och så där, men det är väldigt viktigt att man förklarar att såhär fungerar det, går igenom applikationen, här har vi tänkt att ändra, det här är varför vi gör dom här grejorna. Vi skall

lösa följande saker för den här typen av kund. Veldig, veldig viktig att man går in på varför-biten. Det innebär att om utvecklingsteamet förstår vad det är vi skall lösa, då kan dom mycket lättare hitta en relevant teknisk lösning på problemet. Annars blir det den här situationen att du har detaljspecat exakt vad dom skall göra. Då gör man exakt det och inte mer och stöter man på patrull så lägger man sig ner och dör och väntar på att man skall få nya instruktioner liksom. Det...och den situationen vill jag absolut inte ha utan jag vill ha självgående team. Det innebär att vet dom vart dom skall så kan dom hitta en smart lösning och är det så att dom stöter på problem så vet dom fortfarande vad målet är och kan ta sig dit själva och kanske bara behöver stämma av med den som har designat. Nu stötte vi på den här grejen och vi valde att lösa det så här. För då kommer du fortfarande fram till målet och då säger designern "skitbra" och då fortsätter man och då behöver det inte vara så att det står still mellan specmötena. Dom skall kunna ta sig fram för egen maskin. Det är ungefär som... jag brukar använda försvaret som ett exempel där. Ryska försvaret är ju totalt toppstyrt, så skjuter man chefen så vet inte truppen vad dom skall göra, då går dom hem i allmänhet. Medans svenska försvaret fungerar mer så att varje grupp är målstyrd, så dör chefen så kan fortfarande gruppen lösa uppgiften. Så att är det ett hinder på vägen så vet man ändå att man skall ta kullen t.ex. Då springer man till kullen och tar den. Det är inte värre än så och det är liksom på ungefär samma ledarskapsmetodiker som paketmetodiken är uppbyggd då. Att folk skall ha förståelse för vad dom skall göra men ha frihet att ta sig fram till lösningen på ett optimalt sätt och så jobbar vi hellre med review då. Förstår ni hur jag menar?

M: För att hålla styr på det så att säga.

J: Ja, vi måste ju fortfarande se att gubbarna springer till rätt kulle. Förstår ni vad jag menar?

M: Jaja men.. mmmmm....

B: Ja det var intressant.

M: Ja det var det utan tvekan.

Bilaga 3, Intervjumaterial Adb-kontoret

Transkribering, ADB-kontoret, 2004-03-24

[...]=efterhandskommentarer av intervjuarna

M=Mattias Grytting

B=Björn Olsson

H=ADB-kontoret

T=ADB-kontoret

M: Hur ser företaget på sig själv och vad det gör?

H: Då är ju ADB-kontoret en förvaltning inom Göteborgs stad. Det som skiljer oss ifrån andra förvaltningar inom Göteborgs stad är att vi inte är bidragsfinansierade. Alla andra kommunala förvaltningar får ju bidrag ifrån stadskansliet, men det får ju inte vi, utan vi måste försörja oss själva. Dra in vårt eget kapital, och vi har också krav på oss att lämna pengar till stadskansliet. Man kan säga att på så sätt är vi en kommunal förvaltning som egentligen är i fri konkurrens med alla andra konsultföretag som finns inom branschen. Det som begränsar oss gentemot våra konkurrenter är att vi har ju kommunallagen som styr oss, så vi har ju inte fri konkurrens åt andra hållet. Dom konkurrerar med oss om...och erbjuder it-stöd till Göteborgs stad, kommunens förvaltningar och bolag. Men vi kan inte gå ut på deras marknad fullt ut. Vi är ca. 300 anställda och vi sysslar med allting, från serverdrift, webhotell, övervakning, systemutveckling, de tidiga faserna i verksamhetsutveckling, kravhantering, stöd vid upphandling av it-system, genom hela spektrat. Vi har också många olika plattformar, stordator, Notes, VB, .NET, och Java...ja nästan alla de stora plattformarna har vi, beroende på att vi har så många olika kunder. Vi har...nu kommer Torbjörn [entre]...

M: [repeterar bakgrunden.]

H: [återupptar] Vi har också en blandning av olika system från de stora kommungemensamma systemen som lönesystem, ekonomisystem och system för socialtjänsten. Dom är ju mestadels stordatorbaserade. Lönesystemen har ett Windowsgränssnitt, men det är stordator som är grunden för verksamhetsregler och datbasregler och sådant, som ligger i DB2 och DB1. Sen så har vi web- och Windowsapplikationer, både stora och små och dom är kundspecifika och vissa är för stadsdelsförvaltningen [otydligt]. Det är många typer av....

M: Alltifrån egenutvecklade....

H: Alltifrån egenutvecklade till standardprodukter som är integrerade med nåt annat. Mycket av det vi har är egenutvecklat, där kunden ringer och vill ha ett system och då bygger vi det just från ax till limpa, som du var inne på. Och sen har vi dom här stora systemen, kommungemensamma systemen var ju en gång i tiden nån typ av standardssystem som man gjorde i samarbete mellan Stockholms, Göteborgs och Malmös kommuner, och sen så köpte man loss dom i Göteborg och gjorde anpassningar, så nu har ju de blivit mer egenutvecklade system som är anpassade då. Så där har vi också då...ett sortiment av olika tekniker, miljöer och typer av applikationer. Våra kunder ser också väldigt olika ut, eftersom vi har alla förvaltningar och bolag inom Göteborgs stad. Och sen har vi konsultverksamhet också...där man hyr timme för timme.

M: Och då är det inte nödvändigtvis gentemot kommunala bolag...?

H: Inte nödvändigtvis, men våran främsta marknad är Göteborgs kommun, men vi kan ta jobb utanför kommunen, men då är det nåt regelverk som styr det och det kan jag inte i detalj. Men vi är begränsade, vi kan inte åka ut i hela Sverige och konkurrera med IFS och så. Och sälja våra tjänster på det sättet..... mmm...i stort sett.

T: Vi får göra merförsäljning, som dom kallar det, till andra kunder än vår primära marknad men vi får inte anställa fler resurser för att rikta oss till det som ligger utanför vår primära marknad. Har vi en extern produkt kan vi givetvis erbjuda den till andra också men vi får inte utveckla en produkt för andra. Vi finns ju till för....

M: Så om det är någon extern kund som säger att det ni har utvecklat till en internkund ser så bra ut så att vi vill också ha den så är det ok.

T: Ja det är ok.

H: Om vi inte behöver anställa fler för att kunna genomföra åttagandet då.

M: Men ni marknadsför dom inte aktivt utåt mot andra bolag.

H: Inte på det sättet, det kan vi inte göra.

T: Existerande produkter kan vi ju visa och säga att det här finns att köpa men vi kan inte sälja in nyutveckling på det här.

H: Stämde det med er bild av ADB-kontoret, blir man ju lite nyfiken på?

M: Ja jag hade nog spontant inte trott att... jag hade nog uppfattningen lite som du var inne på här att ni har er bas i dom kommunala bolagen men är egentligen fria att lite mer konkurrera på andra marknader också, mer aktivt än merförsäljningsmässigt.

Det trodde jag, jag var inte medveten om att det fanns så pass skarpa restriktioner på det.

H: Det är ju komunallagen som styr det och det kan ju vara lite jobbigt ibland.

M: Det kan jag tänka mig att det kan vara.

H: Man kan ju uppleva det lite besvärligt, vi är ju utsatta för konkurrens men vi kan inte konkurrera på lika villkor, upplever man när man sitter här tycker jag.

M: Ja det kan jag tänka mig, samtidigt vet jag inte riktigt hur ni ser på det här med... om vi antar nu att ni har god kompetens om systemen som varit förvaltade länge och stordatorsystemet och sånt... Jag vet inte riktigt hur det funkar men jag kan tänka mig att dom externa bolagen, dom andra bolagen som konkurrerar mot er, dom har inte den kompetensen så dom kan inte gå in och erbjuda service och så på dom systemen för dom kan inte dom systemen.

H: Det är ju en av våra styrkor att vi har just den här kunskapen om verksamheten eftersom vi har haft systemen väldigt länge. Vi har ju även kundservicen och supporten och många av dom som jobbar med det kommer ifrån verksamheten en gång i tiden och har väldigt hög kompetens inom det. Och det har ju inte våra konkurrenter.

M: Det är ju en fördel i det läget, kan man tänka sig.

M: Det är ganska viktigt att etablera en gemensam syn på vad en metod är för olika organisationer, det kan vara väldigt mycket och dom kan vara väldigt detaljerade eller väldigt övergripande. Så frågan som vi ställer här är egentligen vad är en metod för er del? Vad innebär den, definiera den litegrann, så att vi talar om samma sak här.

H: Jag ser ju en metod som en sorts processbeskrivning där man ser vilka aktiviteter som skall utföras och i vilken ordning. Lite slarvigt kan man väl säga att det är en gigantisk checklista. "Det här kan du tänka på att göra i den här sekvensen". Bra tips och råd. Ett metodiskt sätt, att man kan upprepa, du kan utföra ett arbete, du kan upprepa det på samma sätt nästa gång i och med att du har följt den här beskrivningen av vad du skall göra.

M: Då blir resultatet också detsamma, så att säga...?

H: Jaa...säkerställa att resultatet blir detsamma.

T: Att metoden även innehåller så att säga mätpunkter, uttalade milstolpar, krav och granskningar. Det är en del av metoden också. Vår metod innefattar att "de här sakerna skall alltid göras". Grillad gris[otydligt]. Det anser jag också är en del av en metod.

M: Vad är en god metod, kontra en metod som inte är bra? Finns det någon skillnad där, kan man definiera en god metod på något sätt?

H: Ett krav är att den skall vara lätt att ta till sig. Den skall vara lättöverskådlig. Är den inte kommer den aldrig att bli använd. Och att den skall vara anpassningsbar. När vi är just inne på anpassning av systemutvecklingsmetoder, det är ju a och o att den är anpassningsbar, till just ditt uppdrag.

T: Speciellt när man har så varierande uppdrag och situationer. Då är det ju ännu viktigare att metoden är anpassningsbar.

B: Ni hamnar ju i en lite konstig sits där, i och med att ni får ta alla uppdrag inom Göteborgs stad. Rätta mig om jag har fel nu, men oavsett om det är i en specifik nisch, ni får ju allt, oavsett om det är ekonomisystem eller....

H: Renhållning, snöröjning, administrativa system, allting, alla olika typer....och därför har vi ju väldigt stora krav på den eller de metoder vi väljer. De skall ju gå att anpassa till olika typer av uppdrag. Storleken på uppdrag, tekniken, storleken på antalet medarbetare....ja...

T: Ja...om det är nyutveckling eller förvaltning, vidareutveckling av redan existerande. Det ställer olika krav på hur det skall se ut...

M: Om man tittar på delarna i en metod, eller vad en metod gör...så...vi satt och ritade på vår kammare lite för att försöka få ett grepp om vad en metod skulle vara. Och då ritade vi oss fram till den här lösningen [visar käck bild]. Vi kallar alltihopa för en process och då sa vi, att för vår del så ingår det som du var inne på förut, lite mera projektstyrningsvariant, där man har väldigt grova drag om vad som händer här...så att man har checkpoints på varnånstans man är jajhjhdfhj[otydligt]. Och inne i den här processen kan man använda sig av en metod eller en metodologi sådär som mer detaljerat talar om att "först skall vi göra det här, sen kommer det" och så...De här delarna kallar vi för aktiviteter, vad det nu kan vara, gör en

analys av befintlig verksamhet eller rita ett klassdiagram, vad det nu kan vara. Vi såg att det fanns ett par olika granularitetsnivåer, den första väldigt övergripande, metodologin från ax till limpa och sen inom den finns mindre bitar då. Är det här nånting som ni ser så att säga, finns det en sån här uppdelning eller är en metod alltid en hel soppa? Är det inte en metod om det inte finns en sån här? [pekar på nåt], en övergripande process eller projektstyrningsdel eller finns det några... så som ni ser på er verksamhet har ni en en en separat projektstyrningsdel och i den stoppar ni er metod som sedan säger att så här skall vi göra systemutvecklingen men projektstyrningen går till så här. Eller är dom invävda i varandra?

H: Dom är invävda i varandra.

M: Det är dom, ok...

H: Vi har en projektstyrningsmodell då, en egenutvecklad modell, den liknar väl dom flesta andra självklart och är inte särskilt märkvärdig, det finns ett antal steg man skall göra och det ser likadant ut över allt. Och den har ju vi integrerat då... vi har ju valt RUP som vår metod för systemutveckling och då har vi integrerat den med det iterativa tänkandet, iterativ planering. Så vi har kopplat ihop dom.

M: Ser ni att det finns för er del begränsningar i att väva ihop metoder på det sättet...? Ta er egenutvecklade projektledningsmetod och lägger ihop den med RUP, vad får det för konsekvenser... om någon nu skulle säga att: "ja jag flyttar mig i processen enligt projektstyrningsmetoden men RUP säger att jag skall göra det här här men det kan jag inte göra för projektstyrningen ser inte ut så". Utan jag skulle villja göra något annat där istället.. ehmm finns det möjlighet att göra det då?

H: Ja.

M: Det gör det? Inom ramarna för RUP?

H: Ja. Utan som vi upplever det så är själva meningen med RUP att man skall anpassa den, du gör fel om du inte anpassar RUP till varje enskilt uppdrag. Du begår ett stort misstag om du inte gör det. Och det gör vi ju[anpassar, får vi förmoda]. Varje nytt uppdrag som vi får så gör vi en processanpassning eller development case som det heter i RUP där vi tittar på varje projekts unika förutsättningar och skräddarsyr metoden till det projektet, plockar bort eller plockar till... byter ut mot något annat.

M: Och då rör vi oss egentligen på det som vi kallar aktivitetsnivå här att man plockar bort vissa delar och lägger till andra delar allteftersom man känner att det behövs.

T: Och även det som eventuellt underliggande... artefakter och när man kanske väljer till saker som existerande system. Eller att man väljer bort saker som inte är intressant att producera just på grund av projektets egenskaper. Det är ju också en del av anpassningen som man gör för projektet.

M: Ja det kan det ju vara... Om vi skall prata lite organisation här då så antar vi att vi har en projektorganisation som skall utföra det här projektet och den här projektorganisationen har förmodligen, och nu spekulerar jag lite grann...dom har en övergripande kunskap om vad systemutvecklingsmetoden innehåller, vilka möjligheter som finns inom den ehhhh och så

vidare... I vilken utsträckning har dom som arbetar i projektet möjlighet att göra dom här justeringarna själva, görs det så att säga innan eller görs det under gång?

H: Det görs under gång. Det första man gör när man startar upp projektet är att man gör en första anpassning av metoden för projektet och det gör man med hela projektgruppen, alla skall vara delaktiga och då har vi ett RUP-team eller vad vi skall kalla det för, som kan ge stöd i dom anpassningarna för dom som inte är så vana att arbeta med RUP då och så att dom vet vad som är lämpligt och inte m.a.p projektets förutsättningar. Sedan gör man ju en bedömning av hur det har varit, tog vi med oss för mycket eller tog vi med oss för lite? Det gör man ju regelbundet då i samband med att vi planerar iterationen och så har vi iterationsutvärdering för varje iteration, så att man tänker till. Hur gick arbetet i denna iterationen? och då resonerar man ju fram sig till hur man har jobbat i processen också. Om den blev för tung eller om det blev för lättviktigt eller om det är något som vi vill ändra. Så gör man anpassningar under gång och vid alla dom tillfällena så är alla i projektet med.

T: Och då kan man även göra justeringar av själva processanpassningen så att säga.

M: Så nästa varv så märker vi att det vi gjorde i förra varvet.... att det gav ingenting.

H: Nej, den behöver vi inte ha med, vi tar den här istället eller vi tar inte med det alls. Vi har inte haft någon nytta av det, vi trodde att vi skulle ha det men vi har inte haft det. Då tar vi inte med den.

M: Då skall vi se... jag var inne på något som jag alldeles glömde av här...

B: Och då tänkte du att jag skulle komma på det? Så där snett in från sidan?

M: Ja jag tänkte ju det men det kanske var lite för mycket begärt?

B: Jag antecknar ju!

H: Han är ju protokollförare, han har annat att göra ju.

M: Jo... Nu är ju inte vi några experter på RUP, jag eller Björn men vad var det som fick er att välja RUP när ni skulle välja metod? Vad var det för argument i den debatten? Om det nu var en debatt?

H: Vi hade en förstudie, det var i samband med en ganska så stor organisationsförändring som vi hade på kontoret för några år sedan, innan dess var vi indelade i sex eller sju olika avdelningar som var indelade i väldigt strikta gränser som jobbade på väldigt olika sätt. Ett av målen med organisationsförändringen var att få alla som jobbade med systemutveckling på ADB-kontoret att arbeta på samma sätt oavsett vilken avdelning du tidigare hade tillhört. Att få ett gemensamt arbetssätt, att arbeta enligt en gemensam metod...ehhh...för att det här skall funka med den nya uppdelningen och att man skall utnyttja resurserna mer och då hade vi ju ett antal olika val, det fanns egenutvecklade metoder på kontoret också... metoder...alltså flera olika. Vissa hade börjat nosa lite på RUP, men väldigt väldigt lite. Och ytterligare andra använde ingen metod utan... ja, metoder man hade i huvudet för att man hade jobbat så här i ett antal år och hade sin egen metod. Och då gjorde vi en förstudie där vi satte oss ner och diskuterade igenom vad som var vår gemensamma... där hade vi en ganska utvecklad egen metod då att välja på eller någon befintlig på marknaden.

Då valde vi bort att ha en egenutvecklad metod för att vi såg att det kräver en ganska stor organisation för att hålla en metod som man själv har utvecklat vid liv. All dokumentation, utbildning och att hålla den up to date o.s.v. Det är nog för stort för ett företag som ADB-kontoret att hålla det här igång och hålla kvalitet på det. Det finns fördelar med att välja en befintlig metod, man får tillgång till nya metoder och man kan uppdatera den och det finns utbildningar och då var det just att RUP är ju en välkänd metod då vet jag inte ens om man hade börjat prata lite om XP möjligtvis men dom var ju inte lika kända som dom är nu i alla fall utan det var ju RUP som var den största metoden inom systemutveckling och att det fanns färdiga utbildningspaket inom alla olika discipliner inom RUP och det finns konsulter om man behöver konsulthjälp så är det lätt att få konsulthjälp som vet vad RUP är och kan jobba enligt RUP och kan falla in i vårt naturliga arbetssätt. Likadant vid nyrekryteringar att när man rekryterar, att man kan säga det att en merit är om du kan RUP så får man in kunnig personal med en gång. Så det var det som gjorde att vi valde RUP då, välkänd metod, lätt att rekrytera och att det är många som har använt den och det har fungerat för dom.

M: Det är ganska intressant när man pratar om det här som en del i ett större skeende i en stor organisationsförändring och då väljer man att säga att “nu passar vi på att titta över det här med vår metodanvändning...” För ibland kan man uppleva, när vi har pratat med folk att om man inför en ny metod så gör man det av flera anledningar och en av dom anledningarna kan vara att välja att strukturera verksamheten efter metoden... Alltså nu kör vi in en ny metod här och då är vi också tvugna att ändra vår verksamhet så att den skall se ut på ett speciellt sätt eller så kan man säga så här, vi har den här organisationen vilken metod passar bäst för den organisation som vi har? Så att man anpassar metoden efter organisationen istället. Fanns det med i debatten också?

H: Man kan väl säga att från början när vi gjorde vår organisationsförändring, definierade man ju upp våra processer i organisationen, inte bara systemutveckling, utan alla, och så gjorde man organisationen lite utifrån det, då. “Vad har vi för processer på ADB-kontoret?” och då var systemutveckling en av dem, av delprocesserna, och det var ju det som gjorde att vi skulle hitta en metod som skulle bli vårt gemensamma arbetssätt för den här processen. Man kan säga att till viss del var det ju processerna som utgjorde underlaget för valet av metod, då.

M: Så utifrån den organisationen som ni valde på systemutvecklingssidan passade RUP in i det ni ville göra sedan...?

Om man jämför till exempel RUP och XP då, Den erfarenheten vi har, om än liten, av både RUP och XP så upplever vi kanske att RUP är den som motsvarar hela alltihopa, alltså det finns delvisa delar i RUP som pratar projektdelar, det finns även sånt som verksamhetsutveckling och sådana bitar, medan XP ligger på en betydligt lägre granularitetsnivå, vi kanske pratar mitt inne här [pekar på vår käcka modell] då nånstans. Hur man verkligen gör, “så här skall vi koda, grabbar”. Fanns det någon debatt om hur mycket täcker det här, hur mycket vill vi ha? När ni jämförde XP och RUP så ser man ganska enkelt att det här är två ganska olika bestar, det här.

H: Vi ville ju ha en heltäckande metod, eftersom vi jobbar från verksamhetsutveckling till drift och produktion, hela linjen, så var det ju en heltäckande metod som var den bästa för oss. Sen kan jag tänka mig att det skulle vara så här, nu har jag inte testat det nångång men, men jag menar, RUP har ju olika discipliner som ni vet och så att säga i implementationsdisciplinen, om man skulle följa XP, jag vet inte om det skulle vara någon motsättning i det, det skulle säkert vara möjligt, att man använder RUP som det övergripande

ramverket, men för implementation lyfter vi ut det som står i RUP och lägger det här och så lyfter vi in XP istället. En sån anpassning skulle man mycket väl kunna göra. Det tror jag.

M: Har ni varit med om den typen av anpassning, att RUP säger att vi skall göra så här men i det här fallet så tror vi att vi kan göra så här istället och lyfter in en annan metod.

H: Exakt så har vi gjort med projektledning... I princip har vi lyft ut det som står om projektmanagement i RUP och lyft in vår egen projektstyrningsmodell Det enda som vi tagit med oss ifrån RUP är iterationsplaneringen och iterationsutvärderingen och kopplat på dom på vår egen projektstyrningsmodell. När det gäller verksamhetsmodellering så kör vi inte heller det enligt RUP, utan vi kör en annan metod och ett annat verktyg än Rationals verktyg. Så det kan man säga att vi har gjort, ja.

M: Varför valde ni att lyfta ut dom delarna. T.ex verksamhetsmodelleringen?

H: Det fanns redan en metod för verksamhetsmodellering som vi var nöjda med, vi hade börjat jobba med våra kunder enligt den metoden och vi var nöjda med den metoden och det verktyget och då kände vi att det inte fanns någon anledning att byta, vi hade något som fungerade väl. När det gäller projektledning så är ju inte det RUPs tyngsta disciplin och då upplevde vi att vår egen var mycket mer heltäckande så då tog vi bara det bästa ifrån RUP.

M: Ibland kan man också tänka sig att när man sätter sig ner och skall välja en metod så finns det ett antal olika saker som kan driva det... omvärldsfaktorer så som vad konkurrenterna använder, ibland kan det gå lite trender i metodutveckling och så där. Att man säger att RUP är bra för det använder alla andra också så vi tar det också eller så kan det vara så att man... och det kan man tänka sig att det gäller lite mer andra organisationer än er, som vänder sig lite mer till den privata marknaden.. alltså om en stor kund kommer och säger att vi vill ha ett system så har kunden redan ett verktyg för att göra systemutveckling där dom säger att ni får använda det här för vi är vana att använda det här och våra personer kan detta. Har ni stött på någon sådan variant?

H: Någon gång har det hänt att våra kunder har sagt att vi skall utföra projektet enligt RUP. Och det har väl också varit en del i varför vi har valt RUP.

T: Ja det måste vi ju kunna leva upp till. Det är en faktor, helt rätt.

H: Sen så har vi ju inte så mycket den typen av kunder som jag upplever det. Som ställer så stora krav på hur vi utvecklar våra system utan snarare att systemet skall stödja verksamheten. Sen hur vi gör är inte deras primära intresse utan dom vill ha ett bra system som stödjer verksamheten... det är det som är deras område. Så vi har ju också olika typer av kunder.

M: Det innebär också lite grann, hur mycket ingår t.ex kunden som sådan i projektgruppen när ni utvecklar ett system?

H: Om man nyutvecklar en applikation så har vi ju ofta med oss kunderna i kravarbetet och det är ju mest dom tidiga faserna då. Sen hänger dom ju givetvis med när det gäller förändring av kraven hela vägen, men tyngdpunkten ligger ju i början när man bygger upp krav på funktionalitet och gränssnitt. Däremot så vet jag inte om man kan säga att dom ingår i projektgruppen på det sättet. Dom fungerar mer som en referensgrupp, dom är ju med och tar hand om kraven.

T: Man plockar in dom som... om man inte har egna verksamhetsspecialister så plockar vi in dom.

M: Det var lite grann det jag tänkte...om man har ett större kommunalt bolag...nu vet jag inte om dom har det men...om vi tänker oss att dom har en egen IT-avdelning, att dom har delvis en liten egen applikationsutveckling som dom klarar av själva vid mindre behov. Är dom involverade i den här processen då? Om dom skall ha ett stort system? Hur smälter dom in i arbetet då?

H: Dom är med som kravställare... däremot... tror jag.... inte att....

T: I enstaka fall ... som dom ... där kunden har...sen har vi haft fall där kunden har haft inne någon extern konsult också där man jobbar ihop med dom så att säga. Men kanske inte så ofta någon från kundens egen verksamhet. Och då har det ju vart t.ex att vi skall jobba enligt RUP och så har någon extern konsult som dom redan har ett samarbete med gjort vissa delar av projektet och vi har gjort andra delar i projektet. Så det har ju hänt.

M: Och det har fungerat också, eftersom man har arbetat enligt RUP så har man vetat vad man har fått därifrån eller?

H, T: Mmmm, ja.

T: Det har fungerat bra.

H: Mmmm. Då vet man ungefär vad som gäller. Sen är det ju som du sa, många av våra kunder har ju inte den typen av it-enhet, inte den att syssla med it-utveckling på det sättet, och därför är det inte så vanligt förekommande. Men just kravställare...dom..det är ju viktigt, det vill ju även att dom skall vara med i kravarbetet.

M: Kanske till och med i verksamhetsmodellering...?

H: Där också. Visst. Från verksamhetsmodellering in i krav och sen är dom ju med i acceptanstester och sånt där också.

M: Ehh....hur ser ni på det här med....passar vissa metoder bättre för anpassning än andra metoder gör och isåfall, varför gör dom det? Vi har försökt få ett helhetsgrepp över metodfloran och då har vi sett att det finns metoder som är så kallat prescriptive, där man säger att "dom här stegen skall vi genomföra, från a,b,c till e. Om man gör det så får man det här resultatet. Om man inte gör det kan vi inte vara ansvariga för vilket resultat det blir." Den typen av metod kallas för prescriptive. Sen finns det en annan variant som kallas för contingency-teorin, där man säger att "vi tycker att anpassning är viktigt och då ser vi till att våra projektmedarbetare eller vad det nu kan vara, har ett antal metoder i sin verktygslåda". Så väljer dom metod utefter hur projektet ser ut, alltså en hel metod. Och så finns det en annan variant som kallas för method-engineering som säger så här: "vi har massa små delar av metoder, s.k method chunks", och sen är det någon eller några som sätter sig ner och tittar i den här lådan när vi skall göra ett projekt och så säger dom: "den här biten kan vi tänka oss, 'plopp', den här biten kan vi tänka oss"...och så matar man ihop en metod just för den här delen. Dom här metoderna som kallas för "out-of-the-box", som man köper som ett paket, till exempel RUP, dom tenderar att vara av typen prescriptive. "Om ni köper det här paketet så gör ni från steg a till steg b till steg c och sen så blir det färdigt och då blir det det här

resultatet”. Och så finns det en annan variant som också är ett paket som man köper men är av en annan typ och heter DSDM, dynamisk systemutvecklingsmetod. Den bygger litegrann på att man har olika små delar, så där. Har ni jobbat med metoder som är olika...nu har ni ju RUP, köpt som ett paket. Tycker ni att det faktum att det är köpt som en out-of-the-box-metodologi begränsar anpassningen i den metoden?

H: Nej. Jag upplever att...vi använder ju RUP nästan som “bygg din egen metod”. “Vi tar lite sånt och lite sånt”...så den har ju...det finns mycket att välja på. Men vi upplever absolut inte att vi måste ta allt. Och vi har ju också, som vi ar sagt innan, bytt ut vissa av byggklssarna i RUP och satt in andra...och ser att det kan funka också. Jag har inte jobbat med någon av de här andra, DSDM eller...jag har inte jobbat med XP heller så jag har inte...jag har mera jobbat efter min egen metod innan RUP kom in i mitt liv, jag har ingen sån erfarenhet.

T: Jag har mest jobbat med RUP också. Jag upplever inte heller den som prescriptive utan snarare att det är det där med method chunks, då...här har vi en metod som skall kunna täcka allting, från utveckling gav realtidssystem...om vi tittar inom den disciplinen som jag är mest verksam inom, analys och design, vi utvecklar inga realtidssystem, då har man ju redan i det skedet som man gör processanpassningen för...sdfhhjhsdfkj[otydligt] för hela organisationen först, och redan i det skedet plockar vi ju bort dom delarna. Skulle det någon gång vara så att vi skulle utveckla [realtidssystem] så kan vi ju plocka in det, som en första anpassning, “det här tar vi bort, det här tar vi med”. Metoden är ju också...den är ju gjord för att klara av dom krav som amerikanska armén ställler som är rigorösa. Då måste man ju ha vett nog att inse att den nivån av formalitet behöver man ju inte alltid ha...och speciellt inte i en organisation av vårans storlek, då. Beroende på vilken verksamhet det är.

M: Det finns, så att säga, utrymme för anpassning så att det räcker i den här typen av metod?

T: Ja...så upplever jag det.

H: Ja...det tycker jag. Vi har inte stött på några sådana begränsningar när vi har jobbat med den.

M: Det är såpass ja...

T: Även dom metoderna dom har jobbat mycket med... producera ingenting som inte ger något gör inte det... Då har man gjort fel. Det är till för att anpassas.

H: Och det är ju konsulter från Rational som har gett oss det rådet, så jag menar att dom tycker ju det. Dom hör ju också det ibland när folk tycker att RUP är för stort och det är så dokumentdrivet och det är för tungt och det passar inte för oss och så vidare. Vi har haft alla dom här diskussionerna, vi har hållit på med ett RUP-införande i två år. Det är ju ett missförstånd och det kan man ju tycka vad man vill om, men det är ju ingen som säger att... jag har ju t.o.m hört folk säga att om man väljer bort något, om man gör en processanpassning så är det RUP då? Om du inte väljer bort något då gör du fel, det är så man skall tänka istället. Och det säger ju dom också som vi har haft kontakt med på Rational när vi har diskuterat det här att... anpassa.... ger det dig något så ta med det annars skall det inte med. Det är grundregeln.

M: Så här lite grann vid sidan av.. den här anpassningen eller införandet av RUP är ju intressant, det är ganska stort.

H: Hehe...ja, det kan man säga.

M: Hur har det fungerat när man skall så att säga rulla ut en sådan här process i en organisation jag kan tänka mig att det blir... innan omorganisationen så nämnde du att vissa avdelningar kanske inte hade en metod som dom var medveta om, även om dom hade den internaliserad för sig själva och hade jobbat med den i väldigt många år och dom kunde den utan och innan och visste precis hur dom skulle göra. Hur fungerar den övergången när man säger att “nu skall vi göra på det här sättet” och så dumpar man RUP i knäet på folk?

H: Men det gjorde vi inte, vi dumpade ingenting i knäet på folk, utan vi tar det här bit för bit det måste man göra... Först jobbade vi väldigt mycket med planeringen med hur det här skulle gå till. Vi har ju ändå 300 anställda och av dom är det 90 som arbetar med systemutveckling och sen kommer det till yttrare 20-30 som jobbar med verksamhetsutveckling, projektledning och den typen av aktiviteter. Hur gör vi nu detta på bästa sätt, köper vi en sitelicens av RUP och säger att nu är det detta som gäller och nu får ni köra efter det? Nej det gör vi INTE för vad var målet med vårt RUP-införande? Jo det var ju att alla skulle ha ett gemensamt arbetssätt och har du RUP online på din pc så betyder ju inte det att vi har ett gemensamt arbetssätt för RUP online är en verktyglåda och du väljer ju vilka verktyg du skall ha för ditt jobb och då kan man ju helt plötsligt ha 20 olika varianter av hur vi tolkar hur man skall göra användningsfall och hur man skall skriva iterationsplaner..[BETONGBORR I BAKGRUNDEN]

M: Betongborr...

H: [Skratt] Ja.... Så det vi gjorde var att vi gick först en kurs i hur man gör RUP-införande och så lyssnade vi på läraren där och sen tog vi konsult hjälp genom Rational. Vi hade lite olika ideer om hur vi ville göra och så fick vi dom reviderade då, och det var ju att börja med ett antal projekt som representerade verksamheten och inte börja med hela RUP på en gång. Så vi gjorde en utvärdering av organisationen och såg vart har vi våra största förbättringsområden, inom vilka discipliner är det? Och utifrån det så valde vi då att börja med...och nu innebär det inte att alla dom här disciplinerna är förbättringsområden, men jag tror att test var ett sådant område, så då valde vi test och kravhantering, analys och design och började med dom här fyra disciplinerna. Ändringshantering hade vi bra koll på redan så det behövde vi inte ta först, så då valde vi tre pilotprojekt. Så satte vi upp en organisation där vi var 8 personer som fick representera dom här olika disciplinerna och tillsammans med externa konsulter från Rational i början eftersom vi inte hade all kompetens själva, gav vi stöd till dom här tre projekten och drog erfarenheter av det här och byggde upp vår första anpassning av RUP för ADB-kontoret. Som Tobbe sa, att vi gör inga realtidssystem det kan vi plocka bort. Vad är det vi tycker är minsta gemensamma nämnare för ADB-kontoret är det något som vi tycker att vi måste ha med alltid och det är speciellt inom projektledningsdisciplinen kan det ju vara lite måsten att här har vi inget val vi måste göra uppföljning o.s.v. Sen när dom tre pilotprojekten var klara så hade vi ju gjort vår första etapp och då utvärderade vi den och tillsammans med ledningen tog vi beslut om att “nu har vi testat RUP och vi tycker att det är värt att fortsätta med den”. På den vägen är det då...nu börjar vi ta in ändringshanteringsdisciplinen...plocka med implementation har vi framför oss nu då...att se hur vi kan integrera det. Och vi har kört nu i sju, åtta, nio, tio projekt enligt den här modellen som vi har gjort med våra piloter då, så att dom får stöd från ett RUP-team, och vi har ju byggt upp mer erfarenhet själva, så nu har vi nästan ingen mentor med oss. Sen genomför vi granskningar med externa mentorer för att säkerställa att vi har gjort på rätt sätt och så. Så så gör vi, vi tar det bit för bit. Vi dumpar inget

i knät på folk utan vi betar oss in i organisationen under en längre period. Vi tror att det är det som är hållbart i längden. Det tar längre tid, men det här handlar ju om att förändra människors arbetssätt och det gör man ju inte genom att installera RUP Online på deras pc.

M: Det bör ju också öka acceptansen för RUP ganska kraftigt.

H: Det som är nackdelen är ju dom som inte berörs i början, dom som inte har varit med i våra RUP-piloter. Vi kan ju omöjligen nå ut till alla....jaa...det kan vi ju, om vi hade varit sju gånger så många som vi är, men det är ju en kostnadsfråga för organisationen också. Dyrt att göra ett RUP-införande. Det är ju inte licenskostnaden, det är ju att förändra arbetssättet som kostar, det tar ju tid och då kan vi ju inte nå ut till alla inom ett år, det går ju inte, och då är det vissa som får vänta, både två och tre år innan det kommer till dom, och då kan man bli lite otålig och tänka "det här med RUP-införandet som ni håller på med, eller någon håller på med, dom andra håller på med, när skall jag få hålla på med det också, då?".

B: Om man tänker...har ADB-kontoret blivit påtvingade några direkta systemutvecklingsmetoder någon gång? Vi pratade lite om att vissa kunder ställer krav, men ni är inte i den situationen att ni blir uppköpta till exempel. När vi pratade med IFS så hade dom t.ex ett joint-venture-bolag med BA Systems i England. Där var det en ganska så intressant situation, för där är det ju två olika företagskulturer och så skall man utveckla en gemensam produkt och vilket är det som gäller?...men den typen av samarbete eller så det.....

H: Neej....

T: Nej. Inte som jag känner till.

H: Ofta så gör vi ju det vi gör själva. Det är inte så ofta vi samarbetar med externa konsultföretag och då har det varit RUP som har gällt för dom också. Det har inte varit några problem.

M: Då har det fungerat som minsta gemensamma nämnare, helt enkelt...?

H: Jaa...precis. Och...nej...påtvingade kan man inte säga att vi har varit.

T: Inte så länge jag har varit här....ADB-kontoret har ju en lång historia.

H: Ja...det kan hända att det har varit nåt längre tillbaka i tiden.

B: När vi satt och konstruerade den här [pekar på vår käcka modell] så tänkte vi lite på vad det var som drev anpassning på olika nivåer. Om man ser en strategisk nivå, som består av ledning och så vidare, en taktisk nivå, som består av mellanledschefer, projektledare, och den operativa nivån där folk sitter och knackar kod och så vidare...så hade vi lite teorier om vad det var som drev på varje nivå.

M: Vi sa såhär: när det gäller den strategiska nivån så var det sådana saker som omvärldsfaktorer – vad gör våra konkurrenter?, hur ser marknaden ut?...den typen av faktorer finns på den strategiska nivån. När det gäller den taktiska nivån så är det saker som intern organisation, som vi har varit inne på litegrann, man kan anpassa metoden efter organisationen eller tvärtom. På den operativa nivån så är det egentligen den enskilde utvecklaren som, enligt den metoden man arbetar med har klart för sig: "det här skall jag

göra” men metoden specificerar inte t.ex. hur man skall göra det, utan den enskilde utvecklaren eller teamet eller vad vi har för enhet här, har full frihet att “ok vår produkt skall vara den här, vi bestämmer att nu gör vi på detta sättet för att ta fram den här”... och då är egentligen frågan hur djupt går RUP i detta? Finns den friheten på den operativa nivån att säga så här...”ja RUP säger att vi skall göra så här nu i den här fasen men RUP talar inte om hur vi skall göra det” eller har man egna moduler i RUP som säger att “här skall vi plocka fram ett klassdiagram tex men RUP säger inte hur vi skall göra det utan vi kan välja det själv” eller får era medarbetare lov att välja det själva så länge dom producerar ett klassdiagram så är det helt ok, finns det en sådan typ av anpassning?

T: Inom analys och design så när man gör användarfsanalys eller om man gör realiseringar eller så så finns det ganska detaljerat specificerat i vilka steg man skall göra vad, givetvis kan man göra anpassningar av detta men det finns ganska detaljerat om man tittar på tex implementationsdisciplinen så kanske man stannar vid att säga att det skall finnas riktlinjer för att ... alltså programmeringsriktlinjer som talar om hur kodstandarden skall vara med namngivning och kommentering, allt sådant. Men dom säger inte hur den skall se ut, utan då kanske vi pekar på en egen kodstandard som är aktuell för den tekniken. Så där stannar man ju på lite högre nivå kan man säga så där finns ju utrymme för den typen av anpassning.

H: Man kan också säga att RUP säger ju att du skall göra ett klassdiagram men det är inget som hindrar att du gör ditt klassdiagram på blädderblock så att säga, det utrymmet lämnar ju RUP, dom rekommenderar ju att man har ett klassdiagram eller vad det nu... eller dom rekommenderar att man tar fram en prototyp men om det är en Powerpoint-presentation eller ritar på tavlan eller om du tar kort och gör bildmontage eller vad du gör, det säger ju inte RUP men däremot så kan man säga att vår organisation har som mål att om du gör ett klassdiagram på det här gemensamma arbetssättet som vi ju har som mål att följa. Att om du gör ett klassdiagram så gör du det i [Rational]Rose och då finns det en mall för det. Om du gör en användarfspecificering så gör du det som ett Word-dokument och då finns det en mall för det... men det är ju inte RUP som säger det utan det är ADB-kontoret som säger det för att vi vill ha ett gemensamt arbetssätt, för om du jobbar som systemanalytiker i ett projekt och tar fram kraven så skall du kunna gå in i ett annat team eller ett annat projekt och jobba med krav utan att behöva säga “ja hur gör ni här då?” och behöva lära om, utan man kan flytta sig....

B: Var det så innan... ni pratade om att ni hade sex stycken avdelningar?

H: Då var det ju inte så viktigt utan man gjorde kanske användarfspecificeringar och dom såg väldigt olika ut.

B: Så det fanns ett problem att gå mellan olika avdelningar.

H: Vi gjorde inte det så mycket heller på grund av att det var inte den typen av organisation där man behövde flytta sig så mycket mellan verksamheterna utan vi hade en ekonomiavdelning som jobbade med ekonomisystem och då jobbade man där och sen hade vi pa-avdelningen som jobbade med pa-system. Men med den här nya organisationen som kräver mycket mer flexibilitet av personalen och då är det här ju en förutsättning.

M: Så då är ju det egentligen som ni har gjort är att komplettera RUP där ni ser att det finns anledning att göra det, som hanteringen av vissa av aktiviteterna t.ex.

T: Sen dom faktorerna som du pratade om på taktisk och operativ nivå... en sak som är intressant är ju t.ex personoberoende på operativ och taktisk nivå när man skall...[BETONGBORR]...när man sitter som utvecklingschef och skall låna in resurser så är du också intresserad av att ha en gemensam systemutvecklingsmetod där bemanningen sker genom roller istället för en namngiven person då. [BETONGBORR] Arbetar man då på samma sätt så ökar det ju möjligheterna att plocka in någon annan och göra samma jobb. Så det är ett faktum som spelar in på bägge dom nivåerna.

H: Det är faktiskt en förändring som har skett nu, upplever jag, under dom här två åren som vi har hållt på att införa RUP, att när man förut bokade Kalle och Anna så bokar man nu systemarkitekt och en testare.

M: Så det är mer eller mindre egalit vems av dom det är som dyker upp, så att säga? Man vet att dom kan systemet, dom vet vad dom skall göra i olika...

H: Vi är inte ända framme riktigt än, men man känner att det är en vridning åt det hållet.

T: Det är något som upplevs som positivt på flera nivåer.

M: Det ökar ju flexibiliteten högst väsentligt.

M: Tänkte litegrann också...på en väldigt övergripande nivå, vi satt faktiskt och diskuterade det här idag, jag och Björt, man kallar...det finns ett uttryck som egentligen heter Systems Development Life Cycle, där man har en syn på hur ett systems livscykel ser ut, alltså, alltifrån design, implementering, förvaltning av ett system, och kanske till och med avveckling av ett system. Det finns ett antal sådana modeller, en av dem heter just SDLC och en annan heter vattenfallsmodellen, den har ni säkert hört talas om eller så. Innehåller RUP någon sådan, som är verkligen övergripande, som talar om hela rasket, förvaltning och så?...ofta är det ju så att en metod stannar vid implementering eller så och sen är man färdig.

H: RUP täcker ju ända fram till driftsättning, från början tog man ju fram RUP som en metod för utvecklingsprojekt. Nu har man väl börjat försöka täcka in det här med förvaltning och...hela systemets livstid, men dom är ju inte framme där riktigt än, tyngdpunkten ligger ju på utvecklingen, men däremot ser man med dom nya versionerna att dom kommer med förslag om hur man gör lämpliga processanpassningar och fasplaner för förvaltningsuppdrag och så. Det fanns ju inte för några år sen, den informationen fanns ju inte att tillgå då. Dom börjar ju försöka täcka in mer och mer.

M: Innebär det då också...jag antar att ni har en hel del system som är på förvaltnings-och kanske till och med avvecklingsnivå. Då innebär det att ni har andra rutiner för hur ni sköter det jobbet om inte RUP täcker in den delen.

H: Vi försöker få RUP att täcka in den delen, det är en del i RUP-införandet i år, det är ett av dom uppdragen vi kommer att jobba med, att se hur RUP kan anpassas till förvaltning. Ännu har vi inte kommit till hur RUP passar till avveckling, men det är förvaltning som är temat för 2004.

M: Då innebär det att ni egentligen gör...tar de beståndsdelarna som ni har i RUP idag och använder dom, flyttar över dom i en förvaltningsdel eller?

H: Vi ser hur dom passar eller om vi behöver göra en specialanpassning, vilket vi förmodligen behöver göra, en specialrekommendation för förvaltning. Vi har gjort en första grov anpassning för kontoren och den kan vi säkert inte bara ta och lämna över på förvaltning, "nu kör vi denna", utan nu måste vi skaffa oss erfarenheter av hur man jobbar med förvaltning med RUP, och då växer det fram en ny ADB-kontorsrekommendation för hur man skall använda RUP i förvaltning. Men även om vi har en ADB-kontorsrekommendation, så anpassar man ändå den till varje projekt, för varje förvaltningsuppdrag. Vi har inte jobbat så mycket med RUP i förvaltning ännu, som sagt har vi börjat med det under 2004. Nu tänker vi gå in...följa med de nyutvecklingsprojekten som vi gett stöd under 2003 över i förvaltning.

M: Det är ju faktiskt klart intressant, då. Har ni hjälp av konsulter ifrån Rational när ni gör den anpassningen eller gör ni den helt själva?

H: Vi kommer säkert att köpa in några timmar, men nu har vi jobbat med RUP så länge här så...vi har byggt upp ganska mycket kompetens. [BETONGBORR]

M: Annars borde det ju vara rätt intressant också för deras del att vara med här...
Det är kanske inte alltid man får tillfälle att vara med i sådana situationer för deras egen....

H: Dom har ju jobbat litegrann med det...Rationals konsulter har jobbat litegrann med det på Volvo, så vi hade en eftermiddag med en av deras konsulter där vi diskuterade lite hur de hade gjort där. Då fick vi lite inspiration av honom. Men det är inte det dom har jobbat mest med, så det är intressant för dom också...

Bilaga 4, Intervjumaterial Wm-data

Transkribering, WM-data, 2004-04-01

[...]=efterhandskommentarer av intervjuarna

M=Mattias Grytting

B=Björn Olsson

L=Intenia

M: Hur vill du beskriva företaget och vad det gör?

L: [Visar Powerpoint-presentation]Varför jag har med den här bilden, det är för att det inte är världens trendigaste killar som står där, utan dom heter faktiskt Thord Wilkne och Hans Mellström. Kortet är nog taget ungefär i början på 80-talet. Det är grundarna av WM-data. Dom startade det här bolaget 1969, den första april, det är väldigt precis på en årsdag idag. Jubileum kan man säga att det är. [Räknar halvhögt] 35 år blir det. WM-data är ett IT-konsultföretag. Det har inte undgått er.

B: Nej.

L: WM-data jobbar med systemutveckling, med applikationer, utveckling av applikationer, leverans av applikationer. IT-drift. Det är 3 stora ben, kan man säga. Tre värdeord som förklarar en del om hur WM-data beter sig är: närhet, pålitlighet och resultat. Kan uppfattas som klyschor, men det är det inte. Närhet, menar vi, dels socialt d.v.s förstå kunden, vad är det egentligen man vill ha, dels geografiskt, eftersom vi finns på en himla massa platser i norden. Och WM-data är ett nordiskt företag, inte ett globalt företag. Pålitlighet, ni känner till, IT-branschen är ju känd för att tyvärr misslyckas med en del saker. WM-data är en mycket stark leverantör på IT-konsultsidan, vi är den ledande, eller en av två ledande egentligen, IT-konsulterna i norden. Och levererar kvalitet minst i nivå med förväntan, det är rätt viktigt. Det där bygger rätt mycket egentligen på att man måste ha rätt bra koll på kravbild och kunden. Att ta ansvar för det vi levererar, d.v.s supporta, kunna vidareutveckla, för att hänga med över lång tid. Också väldigt viktigt. Resultat: när man säljer såna här tjänster, så är det faktiskt resultatet som är det viktiga. Att komma till skott. Att ha hög professionalism, hög serviceanda. När vi kommer in på den punkten, kommer vi in en hel del på det vi skall diskutera idag. Metodik och sätt att hantera systemutvecklingsprocess, applikationsimplementation. Nyttå för kund. Det låter som en självklarhet då, men det är väldigt viktigt att trycka på den, tycker jag.

[Plockar fram bild som visar milstolpar i WM-datas "utveckling"]

En viktig punkt är starten. En annan viktig är börsintroduktionen, som skedde precis i rätt ögonblick, om man ville expandera. Precis i starten på en högkonjunktur. Mycket förvärvsledd expansion, bland annat ett köp av det som hette då Well förr i tiden, en väldigt stor affär egentligen, med mycket kontor, mycket personal. Även med ett köp av nåt som heter Caran, som står för konstruktion, mycket mot fordonsindustrin. Sen vart det första mars 2000 ungefär, och då blev det kris. Så även i WM-data. Kris är väl fel att säga, man vad man gjorde var att man renodlade verksamheten, man tyckte att WM-data var lite för spretigt egentligen. Höll på med för mycket egentligen. Efter att liksom ha växt ut så drog man egentligen ner till att faktiskt jobba med de tjänster vi verkligen är bäst på: systemutveckling, applikationer, outsourcing, drift. Idag kan vi säga, finns det en annan väsentlig affär här... till exempel, antalet medarbetare är 8000, det beror på ett uppköp i Finland av ett företag som heter Novo som jobbar mot den offentliga sektorn i Finland, det är den senaste stora

businessen som skett i WM-data. Det här visar ju lite grann hur resan ser ut, det har hänt ganska mycket, det är klart att det är mycket folk ut och mycket folk in så att säga. Jag själv har suttit i det här bolaget sedan 1996. Affärsiden är att leverera kundnytta genom kompletta design- och it-relaterade tjänster. Vi vill alltså kunna verka i alla discipliner i en leverans alltså, kravfångst, design, arkitektur, implementation, drift, förvaltning och support. Vår marknad: större företag och organisationer i norden. WM-data jobbar inte mot konsumentledet utan framför allt mot dom stora företagen. Vi pratar om topp 60, det är alltså dom 60 största företagen i norden som vi gör mest affärer med, det är där vår absoluta omsättning finns egentligen... Visionen är att vara den ledande nordiska leverantören av design och it-relaterade tjänster, den går alltså ganska väl ihop med affärsiden och om man ser på den senaste utvecklingen så kan man se att WM-data har vunnit, man har klarat sig väldigt bra genom krisen exempelvis. Ser man på Göteborgsmarknanden så är det väldigt många företag som bara har försvunnit, men WM-data har funnits kvar hela tiden och här, i den här regionen, så går WM-data mycket mycket bra och vi rekryterar till och med, nyutexaminerade, vi anställde två för en månad sedan.

M: En av dom är vår kamrat från skolan.

L: Vilken av dom?

M: Andreas Erlandsson.

L: Ja... och vi har ett par stycken till på gång kan man säga. Och det är inte så vanligt i IT-branschen idag. Det är viktigt att notera. Det var ungefär det som jag tänkte säga om WM-data.

B: Det kanske gick förbi mig men vad är din roll på WM-data?

L: Det har vi inte sagt, nej. Jag är konsultchef för en systemutvecklargrupp kan man säga, som jobbar med Microsofts Javaplattform. Jag jobbar också med konceptutveckling, mycket av miljön runt omkring modeller och metoder.

B: Då har vi träffat rätt man med andra ord.

L: Förhoppningsvis. Jag tycker det här är så roligt så jag kan ju dränka er i detta. Men vi skall se vad ni skall få ut av detta.

B: Om vi skulle börja lite enkelt med, vad är en metod egentligen för dig och vad är en god metod isåfall?

L: Det är en bra fråga, ni är ju inte helt obekanta med RUP exempelvis, det är inte vi heller... men jag skall nog börja i den änden ändå faktiskt. RUP har en lång historia som jag inte tänker gå in på, och jag kan den inte heller, inte hela storyn.

B: Det skall du inte avslöja vet du...

L: Jag vill inte kunna hela heller [skratt]. Nej, men det bygger på rätt mycket folk som har haft rätt mycket ideer då, som faktiskt har gått ihop sig och blivit en snygg gruppering, herr Jacobson till exempel. Ivar Jacobson är en viktig drivkraft i detta. WM-data jobbar som sagt med stora stora företag, top 60. Det finns några små företag som till exempel Ericsson och AB

Volvo som har implementerat RUP. Det här är våra kunder då, kanske mer AB Volvo än Ericsson då om man säger så. Det här innebär ju att vi har ju krav på oss, att dels fungera som mentorer och dels kunna leverera projekt med denna metodik och om man bara lite kortfattat skall säga så här då, att i fallet Volvo så har man ju RUP men man har också lagt till sin egen projektmetodik [ritar på tavlan] för att mappa så att säga arbetsformer som man har själv med den här systemutvecklingsmodellen då. Och det bygger mycket på, att jag skulle säga så här att, att här [pekar på tavlan] är det mycket business egentligen och här är det mycket development. Det gäller ju att mixa ihop det här, dels är det så här... man gör en affär, antingen så gör man en affär med en extern kund och man skall leverera någonting, så är det viktigt att man har ett förhållande med den leveransen och då finns det i dom här projektmetodikerna ofta gear[grindar] som säger så här att: "du måste ha uppnått det här för att få gå vidare till det här och när du har gjort det så måste du ha uppnått det här till den här nivån för att få gå vidare dit" och så vidare. Det är ett rätt bra sätt faktiskt att ha kontroll på att vi sköter det här.

B: Mätbarheten ökar markant.

L: Mätbarheten ökar markant.

M: Framför allt är det ju också ett sätt att hantera stora och komplexa processer, om man drar slutsatser från där jag jobbat i 10 år i monteringsfabriken[Volvo Personvagnar]. Om man har ett stort projekt där man skall göra en ny bil till exempel så är det en väldig massa saker som skall kordineras och dom här grindarna då dom blir naturliga stopp att göra det här på. Så i väldigt stora processer så har dom en klar nytta för att strömlinjeforma.

L: Vi jobbar väldigt likartat som våra kunder egentligen, vi har också en egen projektmetodik som vi adderar till. [ritar på tavlan] Vi har en metodik som heter Ratten här på WM-data som också innehåller gear. Så har vi en metodik som vi har i speciella case som heter Accept och Delta som vi kopplar till Ratten. Som mappar väldigt väl med RUP egentligen. Acceptmetoden vet ni kanske inte vad det är?

B: Nej.

L: Det är en metodik för att jobba med processmodellering egentligen. Ericsson är en av uppfinnarna till den. Strängt taget kan man säga att det går ut på att beskriva en process. Så här skulle det kunna se ut [ritar på tavlan]. Det handlar om att ta reda på hur ser eran process ut. Till exempel: hur ser orderprocessen ut för Volvo Lastvagnar när ni säljer en lastbil? Nån lyfter på luren, nån svarar, när man svarar tar man reda på vem det är som ringer, man noterar det här nånstans, man får en första indikation på vad är det kunden vill ha. Man börjar diskutera business, börjar konfigurera en bil. Konfigurera, det vet ju du vad det är...[pekar på Mattias].

M: Ja.

L: Alltså man sätter ihop...jag vill ha extra lång ram, jag vill ha boggie, jag vill ha extra stor tank, jag vill ha luftfjädring och så vill jag ha blåpunktsstereo. Och takbox. Så. Vad händer sen? Jo, vi skall ut och validera det här sen, kanske. Med en fabrik. Kan ni tillverka den här bilen? När kan den vara klar? Sen skall vi leverera bilen till kund, om vi inte skall fakturera. Så beskriver vi, mycket förenklat, dom här stegen. Sen är vi klara. I den processen beskriver vi också vilken systeminteraktion vi har. Vad behöver man här? Jo, man jobbar mot System 1

för att hitta kunden exempelvis, som man pratar med. Går man vidare i processen, vad gör vi då? Jo, då skickar vi lite information till det här systemet, System3. Vad händer sen? Jo, vi får lite extra information från det här systemet, hit. På så vis så har man dels koll på en verksamhet, hur dom jobbar, och dels på vilken systeminteraktion man har.

M: Så det innebär att den här processen, vad man nu kallar den....

L: Accept.

M: Den modellerar både en process, plus att den modellerar dataflödena som tillhör den processen.

L: Ja.

M: Ok.

L: Och det här är ett väldigt tidigt stadiet. Därför att det här är...det är inte ens på use-case-nivå, utan det är väldigt tidigt egentligen.

B: Men det måste det ju vara, för att man skall få ett grepp om verksamheten.

L: Delta-metoden, använder vi senare i processen egentligen. Det är ofta för interaktion med system, liksom, man som användare får en dialog på en skärm, och så får man information där och så trycker man på en knapp. Vad händer sen? Sen slås båda dom här dialogerna ihop och såna här saker. Att forma den. Interaktionen mellan människa och maskin.

M: Vad blir utgången ifrån Delta, blir det interfacebeskrivningar eller?

L: Nej. En interaktionsdesign. Det vill säga..ett storyboard...känner ni till det?

M: Ja.

B: Ja.

L: Ungefär...storyboard. Sen, så börjar vi jobba, kan man säga. Mycket, mycket mer utbyggt än RUP, egentligen. Med analys. Design. Implementation. Dgadjfkjnvhfdejns[otydligt]. Iterativ utveckling. Men, den övergripande metoden, Ratten, handlar ju om WM-datas interaktion med kunden, beställaren. Våra checklistor, hur kontrollerar vi att vi har kvalitet, mitt i projektet, hur jobbar vi med riskeliminering, såna här saker, det ingår ju i mångt och mycket i RUP. Men vi har också adderat till en del saker. En sak som vi adderar till är PRM, "project risk management". En metodik som vi har adderat till då därför att vi tycker att vi måste ha mer kött på benen när det gäller detta, vi måste kontinuerligt ha riskanalyser i projektet för att följa upp... att vi jobbar på rätt sätt, att vi har rätt prestanda, att vi bygger in en korrekt design, att vi inte får prestandaproblem och sådana här saker. Men det ni frågade om egentligen var ju också så här... liksom anpassningen då om vi tar det här paketet då var är anpassningen...? Dels kan man säga att i RUP så finns någonting som heter development case och det är ju faktiskt en inbyggd anpassning i RUP. Är ni bekanta med den?

B: Låt oss säga ytligt, vi har ett begrepp om dom olika disciplinerna och hur dom ser ut och så där men inte på ett detaljerat sätt, nej.

L: Det är egentligen där som man bestämmer hur skall vi använda RUP i det här projektet? Gör man det riktigt bra så säger man , ta bort det, ta bort det, ta bort det och lägg lite extra energi på det. Felet man ofta gör är att man fuskar med detta, vi kör allt. Eller i värsta fall: vi skiter i det, vi kör ingenting. Eller vi kör bara vår egen metodik men vi skiter i det här. Inte alltid så bra.

B: Den här anpassningen av RUP i själva development case, är det någonting som görs en gång när man säger att: “nu skall vi köra RUP grabbar” eller är det någonting som görs inför varje projekt?

L: Bra fråga. Svaret på frågan är faktiskt båda delarna, man gör i regel för företaget ett development case där vi säger att i vår värld så är dom här bitarna oftast mest relevanta. Det vill säga vi paketerar det så att det här är normalfallet i våra projekt.

M: En grundkonfiguration helt enkelt.

L: Ja, men man måste nästan alltid göra det i varje projekt. Därför att ofta är det faktiskt så här att man drunknar i metodiken om man inte gör det.

M: Det finns ett slags uttryck som säger att man gör metodiken för metodikens egen skull. Den dominerar så mycket att man gör den för sakens skull...

L: Och det är faktiskt ett jätteproblem...

B: Jag kan tänka mig också att... nu kör inte IFS RUP men om dom skulle gjort det, att dom har en enda produkt som dom kör anpassning och konsulterar på. Så är det en ganska uniform verksamhet medans WM-data har en mycket större spread på sin verksamhet vilket gör också att det är mycket viktigare att göra en projektanpassning av RUP inför varje projekt. För ett projekt kan vara ett realtidsprojekt ett annat kan vara.... jag vet inte.....

L: ...integration....

B: ...och det ställer också olika krav...

L: Och det kan vara verksamhetsprojekt eller ett infrastrukturprojekt, “nu skall vi stoppa in Windows server 2003 på Borealis, vad gör vi?” Jo det skiljer ju sig kraftigt från ett utvecklingsprojekt men i båda fallen behöver man en metodik. Men, nu skulle jag vilja komma fram till kärnan. Objektmodell och designmodell vet ni vad det är. Och...min erfarenhet av RUP, det är att.....jo, vi har en objektmodell, den här kan man säga att den kommer ju ur det här och det här [pekar]. Analysen där, och kanske delar av detta[pekar]. Objektmodell, UML, samband, här har vi det, här har vi det.....[pekar]. Här har vi lite relationer och lite kardinalitet och så vidare.

M, B: Nu är vi på våran nivå![skratt]

L: Allting dit, där är verksamhetsdriv....man kan prata med marknadschefen om detta[pekar]. “Visst är det så”[pratar med en fiktiv marknadschef]. Så kommer man till nästa steg: IT-projektet då. Det här är alldeles för hög nivå för dom[pekar på objektmodellen]. Då gör dom en designmodell, där man faktiskt kan bryta ner det här mer, detaljera detta tydligare, gör ett

klassdiagram, kanske. Man normaliserar, man kanske delar här[pekar]. Kommer fram till att vi behöver kanske lite mer prylar här för att kunna leva upp till det här. Så att en designmodell är i regel mer detaljerad och mer komplicerad. Så. Jättebra. Livet leker och vi har koll på läget. Nu skall vi utveckla detta, så nu kör vi Rational Rose, till exempel. Så kodgenererar vi. Vi gör alltså färdiga kodstubbar, som bygger på det här. Så, projektet börjar arbeta. Allting är frid och fröjd. Efter en stund så kommer man fram till att det här hänger inte riktigt ihop, vi behöver nog ett snöre där också. Jaha.....och så får vi implementera det, vi gör en ny kodimplementering av de objekten som berörs här nu då. Får lite källkod här borta. Utvecklar den. Håller vi på så en stund. Så går vi tillbaks och tittar i objektmodellen. "Den där hänger inte ihop med den där riktigt"...Så kommer vi senare i projektet...då blir det väldigt bråttom, för nu skall vi nämligen produktionssätta detta på Måndag, och det är Torsdag. Fortfarande är det vissa grejor som inte stämmer riktigt. Då sker det vissa förändringar här[pekar].

B: Som inte reflekteras i....

L: Och så kommer dom inte dit[pekar]. Det här kan jag säga...det här är ett normalfall. I branschen. Och nu är WM-data naturligtvis världsmästare på att lösa detta[ler ironiskt].

M: Ja, naturligtvis[ironiskt tonfall]

L: För...WM-data har nämligen uppfunnit en sak. Vi har faktiskt uppfunnit detta. Vi har uppfunnit någonting som vi kallar för WM-data Development Framework. En sådan här bild har många[pekar]... Nej...vad är det egentligen? Jo....det här är egentligen programvara och plugins till Visual Studio eller Javaplattformar, där vi modellerar detta, i något lämpligt verktyg, Rational Rose eller Visio. Rational Rose kanske man inte har råd med, då kan man ha Visio. Och....jag kan bara säga snabbt vad det går ut på då....Desigmodellen som ligger i Visio, trycker vi på en knapp, säger att generera databas. Den genererar en databas, den ger mig källkod. [skriver på tavlan] Och...det som traditionella miljöer har problem med, det är hur man gör små förändringar, att omgenerera vissa delar. Men det klarar nämligen vårt ramverk. Då frågar man sig såhär: vad ingår i det här ramverket, nu då? Jo...det ingår inte bara verktyg för att kodelgenerera, skapa databaser, utan det finns också andra bitar i detta. Det finns till exempel glossary, code conventions, design guidelines, architect....arkitektur. Det finns alltså ett paket redan, va, som projektet får. "Den här kör ni efter, punkt slut, det finns inga alternativ. Bara kör".

M: Det här blir projektets, vad skall man kalla det, repository, dom har alla sina grejor däri, dom har alla guidelines dom behöver för att sköta.....

L: Det gäller att kvalitetssäkra saker som namnsättning, designpatterns etc. Det här kör vi, vi kör ingenting annat och gör vi något annat så får vi fasan motivera varför vi gör det. Vi har inte råd att låta folk sitta och hacka, det går inte. Då undrar man ju så här, är det inte väldigt tråkigt att vara utvecklare på WM-data, man får ju inte göra det balla.

B: Nu blev jag lite aggressiv, jag skulle villja säga att det är det balla man får lov att göra man får lov att använda mer tid för att verkligen koda affärslogiken istället för att sitta och...

L: Vad överens vi är gubbar.[skratt, alla]

B: Jag har suttit och kodat, det roligaste kodprojektet jag har gjort var i och för sig när jag gick på gymnasiet, teknisk linje och vi fick löda ihop en enchipdator och skriva

operativsystemet till den, men det är ju också... det kan man hålla på med men i det här fallet så är det ju... det får någon annan göra....

L: På det här sättet så säger vi att vi kan leverera modelldriven utveckling för det är nämligen så här... det här ramverket som vi har det är... du kan alltså göra små förändringar i modellen, du kan gå ut och omgenerera delar av en databas. Du får alla databasanrop som bygger på kardinaliteten här, du får alla begrepp som till exempel "find by id" "find order" du får intelligence i utvecklingsmiljön, när jag nu skriver "kundorder." så får jag upp alla attribut automatiskt, jag ser det direkt.

B: Det är samma som Microsoft introducerade i MS Visual Studio 6.0 tror jag eller 5.0 det dom kallar för...

L: Intelligence... fast vi har lagt till modelldelen i den, så vi får med objektmodellen...

B: Det är ju strålande...

L: Det innebär egentligen att vi behöver inte skriva en enda kodrad för persistence, för datalagring.

M: Och det gäller både egentligen... oavsett om vi jobbar i en C++ miljö eller en Javamiljö.

L: Vi jobbar...vi har...vi kör inte den miljön i C++, vi jobbar inte i C++ egentligen. Vi jobbar i C#.

B: I och med att det ligger på pluginnivå i visual studio, eller rätta mig om jag har fel...

L: Nej så är det...

M: Så sköts den genereringen där då...

L: Precis, och det ligger alltså plugins för att ta hand om XMI som är XML-formatet för hur man beskriver modellen... XMI är alltså en anpassning av XML för att beskriva....

B: Det har inte jag stött på i alla fall...

L: XMI är egentligen det som gör att man blir oberoende så att säga. Man kan maskinellt beskriva en objektmodell, den objektmodellen tar vi in i Visual Studio då och utifrån den så genererar vi hela databasen. Vi väljer bara om vi skall ha Oracle eller SQL-server, detta innebär ju också att vi har ju väldigt tajta lager mellan... eller fria lager mellan databas och businesslogik.

B: Den skiktade lösningen blir mycket mer tydlig.

L: Precis. Så det här är inte så lätt va, det är inte så lätt att få 1000 utvecklare eller 2000 utvecklare att få samma bild av hur en skiktad lösning ser ut. Men det här kör vi och alla tycker att detta är underbart. Nu är det precis som du säger, man kan faktiskt mer fokusera på det balla i projektet, man kan jobba mer på hur upplever användaren detta. Den här tråkiga dokumentationsbiten, den gör vi automatiskt i stort sett. Vi får hela det på köpet här. Jag tror jag har en bild här som beskriver det här.

B: Kör ni motsvarande javadoc för C++-koden också?

L: Ja det gör vi för C#-koden då, och det finns också i guidelines då att: "så här gör ni. Punkt slut". Och det är mycket punkt. "In real life model business objects", det här är en liten schematisk modell. Visiomodell, XMI, XML-deskriptorer som egentligen är en del i den här genereringen då för att sedan generera källkod... sen har vi i källkoden flera stycken... här lägger du in businesslogiken. När du gör det här och det här så gör du det här osv. Det här innebär att vi har väldigt lätt för att byta ut folk. "Du...vi måste ha hjälp i Luleå"...."Okej..."...så har man dom här arkitekturen i botten, va. På så vi kan vi faktiskt leverera, liksom, öka tempot med saker och ting, snabbare än alla andra. Vi behöver inte hela tiden diskutera "vilken modell kör ni, hur kör ni, kan Pelle det här, hur gör vi?"

B: Och det tillsammans med att RUP kör stenhårt med att besätta tjänster genom roller så är man ju nästan i mål höll jag på att säga...det var väl lite överdrivet, men....en bra bit på vägen i alla fall.

L: En bra bit på vägen....vi brukar dema....det står demo här[pekar], vi brukar dema detta utifrån en Visiomodell, nu genererar vi koden [tju, tju, tju, tju, tju], skapar databasen, gör en applikation. Det tar en timme, ungefär...eller en halvtimme. Då visar vi hela kedjan, vi gör en web-applikation som bygger på en Visiomodell då. Vi visar kunden, "den här källkoden, den har inte vi skrivit"...."den har vi genererat"...."den slapp du betala för".....så att säga. Ytterst handlar det här om att vara effektiv. Och göra bättre business än alla andra. Ja...det här för oss handlar om att det skall vara enkelt att använda. Det skall inte vara för mycket modell. Det skall inte vara för mycket ramverk, för mycket regelverk, va. Det är fortfarande så att även om man jobbar på det här sättet så har vi möjlighet att faktiskt ändå gå in och skriva SQL eller anropa "stored procedures" i den om vi behöver det, via den här plattformen ändå. Så man säger inte "detta är det sätt du skall köra på, det finns inga alternativ", utan det här är 95% av grejorna. Dom 5% övriga grejorna får vi dokumentera specifikt då, "då gör ni såhär". Den här bilden, som man ofta har i projekt, etablera miljöer, bestämma sig för hur vi skall göra....jag skall bara ta det här[telefonen ringer]

M: Mycket av det får ni gratis menar du....

L: För oss handlar det om varför gör vi det här? Det är naturligtvis jättekul att skriva BOF, [Business Object Framework], men det är ju ytterst det... många, många projekt så sitter man en vecka, två veckor, fem veckor, sex veckor, i stora projekt kan man sitta i månader innan man har koll på, hur gör vi det? Varför gör vi det? Vad skall vi göra? Och det kostar stålar, ni vet, det kostar grymt mycket stålar. Om vi pratar om dom stora företagen alltså, så är det ju ofta involverat personal från deras håll och vårt konsultbolag, andra konsultbolag, det kostar miljoner alltså. Bara i sjön. Vi vill ta bort det och starta här uppe[pekar] i stället, det skall finnas alltså, så kan vi snarare göra anpassningar i det. För det har... jag kan ju säga så här: den här arkitekturen är väldigt väl förankrad så att säga, med Microsoft, med liknande arkitektur. Man har provat den väldigt mycket, man har mönster för hur... vart får man ofta prestandaproblem i sådana här lösningar?, där har vi speciella avsnitt då som behandlar just det. "Gör så här, undvik det här". Det här är tycker jag en viktig del i WM-datas anpassning av RUP egentligen. Och jag kan ju prata om det här i flera veckor.

M: Ja, vi tänkte, när man gör en sådan här uppsats och så då...då skall man sätta sig ner och det skall vara lite akademiskt stuk så att säga.

L: Ja.

M: Och då är dom väldigt inne på att man skall rita sig en egen modell för den teorin man har om hur det här med anpassning och det här med metoder fungerar. Då satt vi och ritade och då kom vi fram till att vi såg på det här med metoder och metodologier på följande sätt, att en del metoder[plockar fram vår modell]...en del metoder finns på den här väldigt övergripande nivån och då kallar vi den för en process egentligen och här ingår det egentligen alltifrån projektstyrning, från start till mål, så här sätter man ihop ett projekt. Så här gör vi när vi avslutar projektet.

B: Den säger vad man skall göra men inte hur. En livscykel helt enkelt.

M: Inuti den här livscykeln finns det ett antal, en eller flera metoder eller metodologier som säger: "hur skall vi nu ta oss igenom den här första fasen med analys? Jo vi skall göra en objektmodell och vi skall göra det...och sen när vi är klara med det så går vi till nästa fas och så skall vi göra det och så". Och inom dom här grejerna, inom varje del av metodologin så är den uppdelad i aktiviteter som säger att: "nu skall vi göra ett klassdiagram" till exempel och "det skall vi göra så här". Jo vi har två mäniskor, kunden är inblandad, vi skall sitta och kasta substantiv till varandra för att se vilka klasser vi skall ha med i vårt diagram. Så det här är den minsta beståndsdelen då, en aktivitet. Och om man nu tittar på det här då, så kan man ju konstatera att... hur skulle du vilja pricka in Ratten, Accept, Delta och RUP i den här kartan, vilken hör till process, vilket är metodologi och vilket är aktivitet? Hur så att säga är dom här sammankopplade?

L: Jag skulle villja säg så här att den här resan som du beskriver här den hanterar vi i Ratten[Lars pekar på process] i WM-data, andra företag kanske Ericsson kör PROPS. AB Volvo har en egen, jag kommer inte ihåg vad den heter men, PSM.

M: Projekt Styrnings Metoden alltså[skratt].

L: Den skulle jag vilja säga tar hand om den här resan[pekar på processen]. Acceptmetoden ligger väldigt tidigt här[pekar på början av metodologiblocket]. Det är alltså hur, kommer vi fram till konsensus med kunden. Om vad som skall göras. Scopet och sånt, va. Börjar man att prata om användningsfall, innan man har liksom bestämt det övergripande scopet, va, så är det väldigt stor risk att man kommer till en kartbild av massa case, funktioner som man....man har ingen överblick i det. Projekt som landar där i början blir väldigt svåra att där....[pekar] plocka in nytt folk. "Kan ni komma in och hjälpa till här?". Sätta dom på banan, vad är det dom håller på med? "Du skall jobba härnånstans" liksom....jättesvårt att komma in. Om man däremot kan ha en övergripande process för att beskriva dom delarna så är det mycket, mycket lättare att föra in folk och få dom att förstå ett sammanhang då. "Okej...jag skall verka i den här delen". Så att...om jag säger såhär....här skulle man kunna säga...generealisera detta....kalla detta för processmodellering istället. Det ligger väldigt tidigt, egentligen ligger det innan use-case-modelleringen i RUP. Sen tycker jag då att den delen som handlar om människa och maskin, alltså interaktion, den tycker jag är tunn i RUP. Man får lite för lite pekpinna egentligen. Där vill vi gärna addera till, men jag vet inte var jag skall säga att den ligger egentligen, jag kan tänka mig att den ligger faktiskt.....här nånstans[pekar].

M: Mmmm....men du ser ändå, så att säga, dom här delarna, dom är egentligen, det är RUP, som tar den biten.

L: Ja.

M: Sen kan det hända att RUPs enskilda discipliner för en sak inte är tillfredsställande och då kan man plocka in en annan. Men det är så att RUP styr det här skeendet i alla fall.

L: Ja, det tycker jag i mångt och mycket att det gör. Men jag kan ju ändå säga...den har ju viss bäring hit[pekar], va. Därför att om jag nu sätter Delta, eller för att generalisera, interaktionsdesign, d.v.s hur skall det här systemet interagera med användarledet?...det börjar man kunna få koll på när man så att säga kommit en bit i analysen, man kanske har gjort inledningen på designen då....designen fortsätter att löpa, men någonstans här är man mogen att ta dom besluten. Fortfarande relativt tidigt i projektet då. Det är här man faktiskt bildar facit litegrann. Så här blir det alltså. Det är då man har nåt konkret att diskutera med användarledet egentligen.

M: Används det på något sätt som du ser det för...kalla det för en extremt tidig prototyp?

J: Ja...det gör det. Det är en del i prototyptänkandet i RUP. Hela RUP bygger ju på att varje iteration ska faktiskt, det behöver inte vara prototyp, men den skall faktiskt generera mervärde för kund. Den skall generera något mervärde för kund. Annars har man inte lyckats egentligen. Man har fört saker och ting framåt, då va. Och det är helt riktigt...den här delen[pekar] är väldigt viktig för interaktionen då, va. I den här delen kommer också ut, om vi generaliserar detta, "output management". Vad är det som skall komma ut ur det här systemet? Extern integration eller rapporter och såna här saker som nån skall läsa. Vad skall det stå i det? Det har ju ofta en jättestor påverkan alltså, på analys och design här alltså. Det brukar vara svårt att kunna ha den här borta, i ett väldigt, väldigt tidigt skede, då kan man på övergripande nivå säger att: "vi behöver det här och det här" men här någonstans[pekar] kan man specia detta tydligare. Men den måste ju finnas långt före här[pekar] va. Om man bara kanske säger såhär...implementation eller construction som det heter, och transition, det här är ju viktiga delar i RUP då kan man säga. Men, så finns det då i RUP och i Ratten, Ratten och RUP har ju också delar här uppe, på den högre nivån[pekar]. Till exempel arkitektur. Alltså den arkitekturen man bestämmer sig för i projektet skall ju bottna i något, den skall ju inte bara... den måste... varför väljer vi en viss arkitektur? Och det handlar ju faktiskt väldigt mycket om "supplementary specifications", vilka krav, förutom rena funktionella krav. Vad är det för andra krav som finns till exempel på språkhantering, att fungera ihop med olika terminaler, stora och små telefoner, etc. Arkitekturen måste ju bottna väldigt starkt i den kravbild man definierar här i början. Men den är ändå väldigt styrande på strategisk nivå. Jag har väl kanske lite svårt att förstå exakt hur den hänger ihop då[vår modell].

M: Vi sa så här egentligen....

L: Men ni har ett streck där[pekar på kopplingen mellan process och strategisk nivå], det glädjer mig.

M: Ja precis, det utgår egentligen från att vi sa att när man talar om en hel process, t.ex ett helt projekt, så tillhör det väldigt ofta den strategiska nivån, vilket innebär att det är t.ex sådana saker som ledningsgrupper i företaget som diskuterar sådana saker som: "hur skall vi sätta samman det här projektet?". "Hur skall vi göra detta?"... Stora bitar. Den taktiska nivån handlar snarare om sådana saker som: "nu har vi startat igång projektet, hur driver vi det på ett bra sätt tro?" Medans den operativa nivån, snarare handlar om: "nu skall vi bygga ett

klassdiagram, hur skall jag göra det?” Det är mer upp till den enskilde utvecklaren, här har du kanske projektledningen[peka på taktisk nivå], här har du kanske den stora dragen, så här gör vi när vi gör ett projekt[strategisk nivå]. Det var egentligen så som vi tänkte när vi gjorde den här modellen. Vi tänkte också så här, att när man anpassar den här processen, om man nu gör det på något sätt, så tas den beslut ofta på en strategisk nivå. Då är det sådana faktorer som driver det beslutet som, omvärldsfaktorer, hur ser marknaden ut? Vad gör våra konkurrenter? När man gör förändringar i den här delen, på den taktiska nivån så är det andra saker som styr den anpassningen, t.ex “vad har vi för fysisk struktur i vårt företag? Är vi divisionaliserade? Har vi en linje och en projekorganisation som vi behöver väva samman? Vad kan det vara?” Och när det gäller anpassning på den operativa nivån, då är det väldigt ofta den enskilde utvecklarens kompetens som styr den. “Jag vet att man kan göra så här och så här, jag väljer att göra så här”. Så tänkte vi på det här och hur man kunde knyta nivåerna till varandra.

L: Jag håller med dig. RUP å andra sidan som är mera systemutvecklingsmetodikerna då, medans Ratten, PROPS eller PSM är mera tycker jag... styr över dels.... riskeliminering, det har man också till stor del här givetvis men det finns också här, affären eller businessen. Kvalitet, gate skriver jag också. Alltså den här övergripande kontrollmomenten, “har jag gjort det här nu?”, på en strategisk nivå. “Vart står vi?” Ett stort problem i systemutvecklingsprojekt är ju att svara på frågan: “hur långt har vi kvar?” Och det är.... jag skulle villja säga så här, att det är väldigt lätt att starta ett projekt men det är väldigt svårt att avsluta. Att avsluta projekt är mycket svårare än att starta det. Därför vill jag gärna...jag bombarderar ofta våra projektledare här med..... “hur tänker du avsluta det? När tänker du avsluta det?” Och den tycker att jag är helt.... “Vad pratar du om? Vi har ju inte börjat ännu”. “Vilka steg har du tänkt att ta? Du skall landa det någonstans också”.

B: Det var en kille från IBM som pratade med oss och frågade: “när påbörjar ni slutdokumentationen?” Då var det ju alla som tyckte att...två veckor innan eller en månad eller så. “Nej, dag ett!” Och det är ju samma sak som du säger här, att projektet kommer att avslutas och därför måste du förbereda dig för det.

L: Och ofta kommunicerar du med kunden, jag menar kunden är ju också helt frågande, “när skall vi börja, vad kommer hända då?” “Vad blir våra roller då?” “Kommer ni bara slänga det i knäet på oss nu?” Det måste man börja planera och fundera på... det är viktigt att er förvaltning här börjar lära sig den här bitarna. Eller också måste vi köpa den här tjänsterna. Vad gör vi? Har vi råd med det eller har vi inte råd? Såna beslut, va måste vi i regel hjälpa till med för att landa ett projekt rätt, då.

M: Om vi skall tala om, litegrann, anpassning lite mera övergripande, så kan vi konstatera att nu så har ni satt den här varianten, så vi säger att RUP är våran grej, vi kombinerar den med Ratten, vi kan Accept och Delta om det blir aktuellt, vi har skapat ett ramverk för modelldriven utveckling, generar kod, alla bitar är på plats och sådär....och då ställer man sig frågan: hur såg det ut innan ni hade det och varför beslutade ni er för att flytta er till där ni är idag? Vad var det som drev den utvecklingen som sa att: “Nej...det här håller inte längre, vi måste göra nåt annat”.

L: Jag har en bild som....den frågan har flera svar, naturligtvis. Som allt annat. [hittar inte bilden] [hittar en gammal taveludd] Ja...hur såg det ut innan? Man kan väldigt grovt generalisera, och säga såhär: förr....för en 10, 15, 20, 30 år sen, så fick WM-data frågor av den här typen: “vi vill ha en duktig cobol-programmerare, vi vill ha en väldigt duktig VB-utvecklare”. Vi får inte den frågorna idag. Den frågorna finns inte idag. Det är ingen som

vill ha en duktig VB-utvecklare. Helt out alltså. Det är ju rätt konstigt. Det är inte så lång tid sedan det verkligen var så att man ville ha en duktig VB-utvecklare. Men den frågan finns inte längre. Det är intressant, för...man vill ha nåt mer. Man vill ha folk som förstår disciplinerna. Man kanske inte uttrycker det så riktigt, men vi säger...RUP...man behöver ha insyn från analys till arkitektur, där är det värsta fallet, då är man verkligen bred. Man kan säga såhär: vi strävar efter att hålla hus härnåstans[pekar]. Vi vill att vårt folk skall hålla till där[pekar] eller hålla till här[pekar]. Implementation, det måste vi kunna. Vi bara utgår ifrån att vi kan det. Det är inte alls det som är det viktiga. Det viktiga är det här[pekar]. Att fånga kravbilderna, för att förstå sig på hur skapar vi en design som är effektiv att implementera, som lever länge, hur sätter vi samman en arkitektur som håller? Dom frågorna, och då är det ju inte alltid intressant att man är en duktig VB-utvecklare längre....Utan vad vi säger, det är att du måste behärska modell. Du måste kunna uttrycka dig i UML. UML.....externa produkter, och utvecklingsmiljö. Kombinerar dom här tre, då är du het. När jag skrev produkter, så menar jag till exempel, var inte så dum att du tror att ni kan göra till exempel en "message broker", köp den. Var inte så dum att du kan skriva världens bästa portal. Köp den. Köp IBM Webserver Portal eller köp Microsoft Content Management Server, köp Streamserver eller vad ni vill. Vi vill gärna....därför bombarderar vi våra konsulter med en produktkunskap kring produkter som vi tycker passar väldigt bra. IBM, Microsoft, t.ex IBM MQ Series, Microsoft BizTalk, Sharepoint, Streamserver plus lite till. Den kunskapen ihop med modelldriven utveckling, kunskap i UML och kunskap i utveckling. Det är den här som gör att du kan ge våra kunder mervärde, för du sätter dig inte och börjar bygga någonting från scratch. Utan du tänker först, "ok, nu gör vi så så så så behöver vi bara bygga det här", det minskar risken och det blir i regel billigare för kunden i slutändan, man är mindre beroende av konsultbolaget och det är faktiskt någonting som vi vill leverera. Alltså ju mer vi försöker binda fast en kund desto större är chansen att vi tappar honom. Ju mer vi bygger en lösning som gör att dom inte är beroende, desto större affär har vi chans att ta.

B: Nu vet jag inte riktigt hur du känner det men... du har mycket bättre insikt i det än vad jag har men när jag har försökt förespråka liknande lösningar för vänner och bekanta... se till att använda utvecklingsmiljöer, ta den hjälpen som du kan få och sen se till att göra affärslogiken jävligt vass istället så får man ofta mothugg... och folk säger att vadå, jag vill ha kontroll på hela processen och nej det genererar dålig kod och det är ursäkterna finns i högvis. Hur går det övertyga...

L: Det är faktiskt gänget som sitter där ute[pekar på Andreas Erlandsson och gänget på rastgården] ett bra exempel på, hela det gänget jobbar alltså med modelldriven utveckling som sitter där ute. Dom solar, det gillar jag inte[ironiskt tonfall].

B: Dom har suttit en halvtimme där också.[flin]

M: Till och med vår kamrat sitter där också ja...

L: Er kamrat också ja, Andreas Erlandsson...

Det här är en mognadsfråga, man måste fatta att så ser businessen ut, man måste förstå det och köpa det. Men gör man inte det då kan man inte vara en kugge i industriell utveckling, det är för mycket hackarmentalitet i det egentligen. Det finns inslag här som kräver så att säga riktiga nördar om ni ursäktar uttrycket. Den delen har vi fortfarande behov av...

B: Jo, men det är dom 5%...

L: Jag skall säga en sak till... Det finns en drivkraft till i detta, kunderna. Vi började jobba med det här, modelldriven utveckling 1997 för energibranschen, där började WM-data utvecklingen av det här ramverket egentligen. Beroende på, stor konkurrens, hög prisrätt, ta affärer. En annan sektor som har drivit oss i detta är bank och finans, väldigt tigt. Bilindustrin, mycket effektiv beställare egentligen och framför allt är dom väldigt bra på att effektivisera sina egna processer. Hur kan man bygga en volvo för 200.000-300.000? Det är fantastiskt billigt egentligen, alltså egentligen, om man ser på vilken enorm utveckling som ligger bakom. Så dom har ju väldigt hög mognadsgrad i detta och förmår sig att ställa bra krav på oss. För att klara av att leverera till dom så förväntas vi leva upp till sådana här saker. En annan finess är offshoreutveckling. Utveckling ofta någonstans, Estland, Kina, Indien, Polen eller någon annanstans. WM-data ser inte detta som ett hot, vi ser det som en möjlighet. Därför att vi tror att dom här rollerna, dom måste vi ha alltså, vi måste ha dom nära kund. Det här måste vi ha väldigt nära verksamheten. Att lägga det någonstans i Långtbortistan, då har man tappat kontrollen, och det känner vi redan. Vi jobbar idag redan väldigt mycket med implementatörer i Polen och Indien faktiskt. Och har egna avtal med både Indien och Polen...eller Indien och Estland snarare. Det är fortfarande i dess linda, men då är det extremt viktigt med den här modelldriven utveckling. Och det som jag kallar för BOF, Business Object Framework. Modell, kommunicera verksamhetens krav, generera kod, okej..nu kan vi börja implementera. Arkitekturen behåller vi här. Det här är...dels...marknaden driver detta. Marknaden driver ju naturligtvis det också då[pekar]. Men det här är viktiga drivkrafter till att dels den här typen av frågor har vi inte längre[bra cobol-programmerare]..och dels...offshoreutvecklingen gör att det är jätteviktigt att ha....det här är ju jättekul, för det är ju nyckeln till att klara av det.

M: Anta nu... vi tittade på... pratade lite om hur det gick från det ena till det andra... när man nu upptäcker det här att nu har marknaden ändrat kraven vi kan inte.... vi får ha en annan modell för att lösa det här så att säga... och då kommer man till något läge då man säger så här att: "ok vi har plockat fram ratten och dom här bitarna så, det är klart men en systemutvecklingsmetod behöver vi". Hur väljer man en sådan? Valde man bland några alternativ? Vad var viktigt för att man skulle så att säga... säga att RUP är bra, det skall vi ha?

L: Jag kan säga så här, att vi har ju som alla andra företag, egna systemutvecklingsmetoder, SUMO heter den faktiskt, och den påminner i mångt och mycket om RUP. Det är bara att RUP skulle jag vilja säga.. det finns mycket kritik om RUP skulle jag vilja säga, och det håller jag med om. Den är mycket... man kan drunkna i den och vi har gjort det valet att våra kunder för det första tror väldigt mycket på denna och jag skulle vilja säga så här, jag skulle kunna tänka mig att vi skulle kunna skippa R och bara köra UP. Unified Process. Därför att RUP, den är väldigt baserad på Rational och det gillar inte vi egentligen. Utan vi vill ha något mer oberoende, därför har vi ett XMI som gör oss mer oberoende av verktygen i gruppen så att vi kan jobba UP med Visio t.ex som modelleringsverktyg. Men det som är viktigt tycker jag det är... i många sammanhang säger man så här... vi jobbar med iterativ... vi jobbar med användningsfall... så gör man användningsfall, alltså usecase som finns i RUP. Och vi tycker att det passar jättebra, det är jag livrädd för alltså då säger man nej till mycket annat.

M: Du menar att man har det som enda...

L: Ja, att man säger att vi använder användningsfall och så har vi Ratten t.ex. Det räcker inte, du har för mycket hål i det. Du tappar det här med arkitektur, riskeliminering, design, modellering. Hela den kedjan med att dom kraven där faktiskt återspeglas hela vägen.

Sen har Rational många verktyg som är väldigt bra, tex för CM, Configuration Management. Hur håller man ordning på ett utvecklingsprojekt där utvecklarna sitter och jobbar med version 3, version 2.7 är deployad i Gent och någon annanstans? Sedan finns det en anpassning som görs för en separat marknad som är den här versionen, hur rattar man det? Det finns ingen människa som kan klara det, man måste ha datorstöd för det, man jobbar med brancher[branches]. En utveckling där man krymper där, dom tilläggen man gör här kan man lyfta ut och lägga in i en annan line. Det är egentligen maskinellt, det är jättesvårt men det går och det är... här ligger andra näringar långt före än vad IT-branschen gör. Till exempel, se på flygindustrin. Eller fordonsindustrin, dom har mycket bättre...eller tradionellt sett har haft mycket bättre koll på detta.

M: Dom har kanske haft behovet under en lite längre tid också.

L: Ja. Egentligen tror jag att IT-branschen har haft behovet lika länge egentligen, men det har varit mycket entreprenörsanda och hej och hå och kul och sådär, va. Och alla vi som har jobbat med VIC 64 då va.....sådär va. Vi vill ju leverera riktiga system till industrin som kan ge dom mervärde till sin verksamhet, va. Mervärde innebär att det skall funka år ett och år två, det skall hålla för uppgraderingar framöver, etc, etc, etc. Många sådana aspekter då va. Läger man in det i ett helhetsperspektiv så tycker vi att RUP är ett jättebra utgångsläge då va. Tittar man exempelvis på det som heter "supplementary specifications", inga direkta funktionella krav, utan annat lull-lull, som man vill ha, som man inte kan formulera i användningsfall. Var tar man hand om det? Det finns 1000 sätt att ta hand om det om man nu bara säger att vi skall köra användningsfall och så kör vi PSM[Volvos projektstyrningsmetodik]. Men i RUP har vi faktiskt...du har mekanismer, dokumentation, mallar och ideer för hur du skall ratta detta. Det finns rollbeskrivningar i RUP exempelvis, dom är 29 stycken eller vad dom nu är, som talar om att du har den rollen, och den rollen och den rollen och du har den och den och den rollen. "Vad skall jag göra här då?" Det är ju bara att läsa. Det står. Det finns en spec på hur du skall göra där, det du ansvarar för. Det där, har alltid varit ett problem i IT-branschen, titlar är inte lika lätta att tyda, som det är i andra branscher. Det råder faktiskt också modellen bot på i viss mån. Våra kunder är ofta väldigt mogna skulle jag vilja säga. Dom är duktiga beställare. Och en del av dem beställer faktiskt saker...dom pratar..."vi behöver hjälp här med processmodellering i ett tidigt skede här innan vi går in i en tidig inception-fas" till exempel. Eller en kravfångstspec. "Vi skall nu gå in i konstruktion och vi har gjort en design"....så man börjar prata litegrann utifrån termerna i RUP, va, och det är våra kunder, det är ett väldigt bra sätt för oss att förhålla oss till kunden då va, visa att vi talar samma språk.

M: Det känns litegrann som om att ni får det lite gratis, då. När man kör RUP, så att säga, så får man en färdig terminologi som även används ute så slipper man ett potentiellt problem, kanske.

L: Ja. Och sen så skall man alltid addera till då va. Och det är där development case kommer in. En metodik lever aldrig enbart, bara metodiken gör inga underverk, va, men det är ett stöd, som hjälper en att göra ett jobb. Men, det finns mycket saker som är medtaget där som man annars skulle få uppfinna själv. Det finns mycket malldokument med rubriksättning, "det här dokumentet skall omfatta dom här bitarna, därför att det kommer vi att behöva i det här skedet", va. Och att sätta sig ner för att försöka hitta sin egen nisch, helt och hållet....njaa...det är inte WM-datas huvudsakliga nisch att hitta på det alltså. Däremot kan vi ta åt oss av vad andra gjort.

M: Händer det vid något tillfälle att, när ni tar ett uppdrag för att utveckla nånting, att den kunden har krav på er att man skall använda en hos kunden redan befintlig metodik, som till exempel inte är RUP?

L: Oh ja.

M: Hur hanterar man en sån situation?

L: Nu kan man säga så här att faktum är att dom som...det är inte så många företag som har en väl utvecklad systemutvecklingsmetod, en konkurrent till RUP. Det finns inte så många egentligen, utan tittar man på stora beställare som Ericsson och Volvo exempelvis, så dom pratar det här språket[RUP]. Sen har vi mindre kunder, och dom kan snarare ha problem att tillämpa RUP, va. Då får vi verkligen vara vaksamma, så att säga, och göra ett rejält development case då, där vi kanske väljer bort rätt mycket av RUP, medvetet då därför att kunden är inte på banan i den processen, dom kan inte den processen överhuvudtaget, va. Då har vi snarare ofta ett helhetsåtagande för den här leveransen... men det finns ju många fall också då vi måste anpassa oss helt och hållet... Nu är det så här att i det här projektet så här och så här och så här. Och då får man utbilda sina konsulter i den här metodiken eller förutsätta att man kan det då och då är det ett konkurrensmedel. Vi har ju flera konsulter som behärskar flera metodiker.

M: Det är intressant, när man läser om det här med anpassning av metodologier så stöter man på en term som kallas för method engineering, den innebär att om man säger så här... man har ett gäng människor som är duktiga på metodiker dom här metoderna monterar vi ner i små, små, små delar och gömmer i skogen...

[skratt, alla]

M: .. nej det gör man inte, men dom monterar ner dom i små, små delar som dom kallar för "method chunks", dom här små bitarna lägger dom i en stor säck och så säger dom det här är vårt arv. Det här är vår säck. Sen kommer ett projekt och så säger man: "ok kunden vill att vi skall göra det här". Så öppnar man säcken och så tittar man vad har vi för ingredienser i vår säck som skulle kunna passa till detta. Så lyfter man ut en chunk och så säger man att det här är den inledande och sen tar man nästa och så snickrar man ihop en metod som är helt dynamisk egentligen. Dom här bitarna är självstände egentligen så dom kan placeras var som helst. Och så syr man ihop en metod av det för varje enskilt projekt. Kan man säga att dom här människorna delvis spelar den rollen i vissa lägen?

L: Jag skulle vilja säga så här att, den säcken har vi inte egentligen och våra kunder har inte råd. Man har inte råd idag att utveckla en metodik för ett projekt i regel. Utan man vill försöka använda industristandard på ett eller annat sätt, UML är en jätteviktig del i RUP och i andra metodiker, så att snarare handlar det om att man har en påse som heter RUP och en påse som heter något annat och så ser man vad kunden vill ha och så tittar man på hur kan man anpassa den här metodiken genom att plocka bort saker och ting och ändå leverera det som kunden vill ha. För vi tror nog att det ofta är ganska farligt att mixa dessa metoder och modeller med varandra hej vilt. Det krävs att vi utbildar vår egen personal, "hur hade vi tänkt oss att det här skulle limma ihop egentligen?" Den resan är en dyr startsträcka egentligen, för oss och för kunden. Så att tipset är ju egentligen att försök att använda i WM-datas fall RUP och gör ett duktigt development case. Försök att verkligen aktivt plocka bort det som inte behövs. Det här kommer inte att kosta mer än det smakar i det här projektet då. Jag förstår vad du menar, men

det är lite det jag menar med att vissa säger att dom använder användningsfall och så kör vi Ratten eller PROPS eller vad vi nu kör. Vi tycker att det blir håll i det för lätt. Man saknar vissa delar då.

M: Har ni upplevt att det t.ex... RUP då, är i några fall inte tillräckligt flexibelt för den situationen som har uppstått för er del, alltså visst kan man köra ett development case men det finns fortfarande spärrar i RUP som säger att vi skulle vilja plocka bort eller plocka till andra delar men då funkar det fan inte..

L: Svar ja. Man kan säga så här, att ofta är en konflikt följande: det är affären som ligger bakom projektet, man säger så här... "WM-data, kan ni vara snälla och leverera det här?" Det kostar X miljoner, har vi offererat det för. Så skall vi köra RUP, det vill kunden och det vill vi. Då konstaterar vi att det är egentligen omöjligt att lägga ett fast pris innan man har gjort elaboration, innan du har gjort elaboration, alltså att... stor del av analys och design egentligen, så har du egentligen inte grepp om vad du skall göra, hur kan vi då gå ut och sätta ett fast pris? Det håller inte för den affärsmodellen och det får man vara vaksam på då och då försöker vi diskutera med våra kunder att vi skulle nog villja rekommendera er att ni faktiskt tar den här resan. En analys, lite design och så fokuserar vi på dom här problemen så att säga. Utifrån det, därifrån kan vi garantera en leveranstid och sätta fast pris, då va. Då har vi så att säga kört RUP och businessen, så att säga mixat ihop dom, men det är ju naturligtvis så här att det förekommer ju att man tar ett fastpris innan man gjort på det här sättet. Det är svårt. Där får man ofta den diffen som du pratade om då.

M: En annan sak som vi tänkte på, framförallt, vi kan ju i i och för sig ta den här....vi pratade med ADB-kontoret då, dom har precis rullat in RUP i sin organisation. Dom har ju ett antal legacy-system som dom behöver förvalta inom kommunen. Äldre system. Då sa dom såhär: "ja...här kan vi se ibland ett problem med RUP därför att den förvaltningsdisciplinen, den del som skall klara ut hur man skall förvalta system, den är inte speciellt väl utbyggd i RUP, ännu". Har ni stött på den typen av problematik också?

L: Ja, absolut. Man kan också säga så här att vi jobbar ju väldigt ofta med legacy-system, alltså gamla system, det finns inga företag som har råd att slänga ut all legacy, va, utan vi skall interagera med dom. Då kan effekten ofta bli så här, att kravställaren är inte kunden utan det är kundens system. Den här informationen finns, det är den du får, det är den du skall jobba med. Kunden kanske säger att: "vi vill ha det här". Men, vadå, det finns ju inte. Hur adderar vi till det på ett snyggt sätt? Ja...det krävs ju en change-request på den förvaltningsorganisationen som äger legacyn, där va. Det måste man ha koll på då. Det är klart att det täcker inte RUP in egentligen. Det tycker inte jag heller. Men är man bara medveten om detta som dels kund och dels leverantör va, så kan man klara det, va.

M: Man löser det ad-hoc, man tittar på situationen och så säger man "vi får göra såhär", helt enkelt.

L: Precis...att vara pragmatisk. Men det du tar upp är intressant va. Det är ju ändå i sådana här konstellationer som en modell skall funka. Det är svårt att få den att funka.

M: Det bevisar ju bara egentligen att det är väldigt svårt att hitta en modell som är så heltäckande att alla är nöjda med den i alla lägen, alltså. Det är inte helt lätt, även om det finns bra försök.

L: ...och det man säger, att systemen skall vara drivna utifrån verksamhetens krav....dom skall inte vara teknikdrivna. Ofta kan faktiskt legacy-system skapa ett teknikberoende. Och göra att det faktiskt är tekniska krav som gör att det blir på ett visst sätt. Det vill ju inte vi gärna köpa, men är det så att man har bestämt sig för att det här skall finnas kvar, va, ja, då måste det påverka slutprodukten då. Är inte kunden med på noterna där, då har man ett problem. Ett kommunikativt problem. "Men, snälla kund, så här är det va". "Skall du åstadkomma det, så måste ni lyckas med det, då måste ni satsa miljoner där". Innan vi kan åstadkomma det ni vill ha.

B: Utan användare och ekonomichefer skulle det bli mycket lättare.

L: Jaa...[skratt]

M: En annan intressant fråga där också var, med utgångspunkt ifrån ADB-kontoret, dom hade haft rätt mycket nytta av, när dom köpt RUP, dom konsulterna som följer med, som hjälper till att implementera RUP och sätta det i företaget, vara med i den här diskussionen, hur man plockar in och ur moduler. Hade ni också den typen av nytta av Rationals människor?

J: Ja...och det kan man säga att vi har haft. Dels kan man säga att vi har gjort så här: vi har ett certifieringsprogram inom WM-data, där vi certifierar ett stort antal WM-datingar i RUP då, och det gör vi med IBM i USA egentligen, direkt, och så köper vi då in enskilda insatser från Rational då när vi så att säga implementerar RUP. Man skall ha klart för sig det att när vi implementerar RUP så gör vi det för att stötta vår kund så att säga och ofta har vi också gemensam låda med kunden där vi så att säga har mentorstöd från Rational då i ett projekt. Men visst, där kommer ju en väldigt viktig kunskap in, t.ex "best practice" då. Hur har andra gjort i det här sammanhanget?, hur gör vi egentligen?, dom frågorna. Istället för att älta dom i ett år så frågar vi dom hur dom har löst det. Det fungerar rätt bra.

M: Dom har kanske en annan överblick då.

L: Visst... även om dom... sen är det ju så här att det är klart att en Rationalkonsult har ju sitt fokus... skall man vara riktigt bedrövlig så kan man säga det att det kan ju finnas tendenser ibland till att man ser att man kanske vill göra någonting för att sälja licenser. Det anser vi att det är produkt och teknikdrivet på ett annat sätt än kravdrivet. Det är en fara och det får man vara vaksam också över. Så vårt bästa försvar är att vara pålästa, så därför certifierar vi ett stort antal konsulter då. Just nu har vi 50 personer som certifierar sig, det sker ju hela tiden... med nya kullar. Se det från skolan så är det ofta så att det är väldigt få egentligen som från utbildningarna som kan det här egentligen...

M: Menar du RUP specifikt eller?

L: Ja. Det är lite konstigt egentligen, speciellt här i Göteborg där det finns en marknad som är ganska tydlig, RUP-driven faktiskt.

B: Jag sitter som ordförande i en sektion i kåren som heter sektionen för It och Informatik som har Informatikinstitutionen och IT-universitetet på Lindholmen under sig. Och vi har försökt att gå ut och pressa institutionerna på att ha... undervisa i verkliga metodologier som faktiskt används, men det går fruktansvärt trögt.

L: Det är rätt intressant vi kan se vissa lokala grejer, Luleå t.ex, kör Rational men det gör man inte i Göteborg. Det tycker jag är lite konstigt.

M: Jag tror också att det finns... av olika anledningar, men jag tror att det delvis beror på att det finns inom den akademiska världen att binda sig hårt mot ett kommersiellt koncept. Jag tror att det är det som ligger bakom den diskussionen i mångt och mycket.

L: Men just vad gäller Rational så har jag full förståelse för det men däremot om jag säger UML inte ens UML är det en del som har läst.

M: Det kör ju vi.

B: Men det tog ett tag.

M: Det tog ett tag innan det dök upp på scenen ja.

L: Det är språket, det är ju det som är språket.

M: Jag tänkte på en annan sak när vi pratade om offshoreutveckling och sådant så inbillar jag mig att det har hänt under en viss period och du visade på skärmen här också att det fanns en expansionsperiod hos WM-data där man köpte andra företag. Hur ser du på det här, att man har en väl satt metod i WM-data, man köper Pelles IT-bolag någonstans och så säger man till Pelle att: "nu skall du utveckla så här i fortsättningen, varsågod". Eller ett gäng indier, "nu får ni fan köra efter den här metoden". Skapar det problem? Hur löser man den integrationen? Säger man åt dom att följa strömmen eller lämnar man flexibilitet åt dom att göra som dom vill?

L: Det bästa sättet är ju att lyckas få ett incitament för att få dom att tycka att: "det här är ju smart det här ju bra, det här är ju kul". Och det har ju vi lyckats med med den här modelldrivna utvecklingen. Där är alltså såväl projektledare som utvecklare är liksom lyriska av det tillägget till UML och modelldriven utveckling. Den sätter liksom pricket över i:et. Den gör det möjligt att vara akademisk och jobba med modellering och få det att funka hela vägen. Man kan till och med säga så här att, är det ett projekt som inte lyckas hålla modellen tigt med implementationen, då kan man säga: "stackars dom verksamhetsanalytikerna" som sitter där och modellerar ihop nåt junk som inte implementeras. Det är faktiskt det som händer. På det här sättet får man ju faktiskt analytikerna att vara mycket mera med i leveransen av slutprodukten också. Så att det sämsta sättet att föra ut det här är ju att piska folk, va. Lägga nåt i knät, "ta det här", utan det är snarare så att i WM-data så har vi en ide om att vi vill hela tiden jobba med återvinning. Vi vill att varje medarbetare skall känna att: "det här är nåt jag har gjort förr och jag har möjlighet att glida med och ta vara på det. Jag kan skydda mig själv från att traska ner i träsket". Genom att faktiskt lyssna på vad andra har gjort. Vi har ju alltså en massa stödsystem för detta också, till exempel har vi en databasweb som handlar om återvinning. Man publicerar alltså till exempel artefakter från ett RUP-projekt i en web, vilket gör att jag kan titta på: "hur ser ett software architecture-dokument ut i det här projektet".

B: Det är som MSDN, fast i RUP istället....

L: Ja..kan man säga. Har man ingen modell, har man inte RUP eller något motsvarande, då har man ju egentligen inte heller en dokumentation, som man vet att: "det här dokumentet är

det. Det här är skrivet i den här fasen eller i den här disciplinen”. Det har man ingen koll på om man inte har en medveten modell, va. Där kommer behovet av att ha en enhetligt implementerad modell, kommer fram som en fördel då. Det här innebär att vi kan gå in och titta då, det heter “experience recovery card”, ERC, ligger högst upp som en favorit i våran internhjälp, då. Där kan man alltså gå in och söka på, till exempel då webprojekt, e-handelsplatser eller vad det nu är, i den här miljön. Och så får man fram då...i vår projektstyrningsmodell Ratten då, så finns det en slutrapport och lite såna här olika dokument, som vi tycker är viktiga. Om man tittar tillbaks på projektet då, “så här körde vi projektet”, organisationen var uppbyggd på det här sättet, så här i efterhand skulle vi kanske gjort såhär och inte gjort såhär, typiskt såna erfarenheter då. Att få med sig den typen av erfarenheter när man går in i ett nytt projekt är ju guld alltså. Det handlar om, för den enskilde, att lyckas, va. Det är ju...om du jobbar på WM-data så är det ju din möjlighet att göra rätt och bli liksom...: “ja han...han Jytting, han. Han var ju med där och där och det gick ju kanonbra”. Det är ju rätt bra att ha det recordet med sig, va. Mot att istället ha: “han var med där...det gick åt skogen, det gick åt skogen. Vad är det för kille?”. Det vill man ju inte hamna i, och det kan vara ett incitament att faktiskt acceptera en metodik, va. Det självdrivna sättet vill vi gärna implementera RUP på. Och våra modeller, då i WM-data. Sen kör vi med nätverk, runtomkring detta. Alltså, människor i varje region, som är...har kontakt med varandra över regionerna så här, omkring dom här projekten. Där man samlas kring just den här plattformen, modelldriven utveckling, där vi också har projektstöd, om man vill etablera ett nytt projekt, så får man lite hjälp härifrån, från dom som jobbar med det här modelldriven utveckling då. “Okej, nu skall vi se här. Sätt upp projektet så här, gör så här, så här och så här”. Det är också väldigt uppskattat, från ett projekt, att få den hjälpen, va. För, jag menar, hur duktig man än är va, så sitter man ändå där va: “hur skall vi göra nu?”.

M: Man kan tolka dokument på lite olika sätt och så där....

L: Mmmm.

M: Det var nog det, skulle jag misstänka.

B: Jaa...jag är också nöjd.

Bilaga 5, Intervjumaterial Intentia

Transkribering, Intentia, 2004-03-31

[...]=efterhandskommentarer av intervjuarna

M=Mattias Grytting

B=Björn Olsson

P=Intentia

T=Intentia

B: Vad gör Intentia...vad är Intentia? Hur skulle ni beskriva Intentia? Är ni konsulter eller säljer ni en egen produkt?

P: Vi är en blandning...allting bygger på att vi utvecklar en egen produkt, en egen programvara i botten. Sen...konsultar vi runt det också då. Men egentligen bara runt det då. Inga lösa konsultuppdrag då, utan det är...i botten finns Movex då, som är systemet...sen kan det ju vara avläggare eller liksom små tilläggsgrejor eller sådär eller hopkopplingar med andra system eller andra delsystem som kunden har då, men det är alltid med Movex inblandat, det är det som är grunden. Och det är ett helt eget system som vi har som täcker det mesta, eller allt vad man behöver för administrativ databehandling och för...ja....tillverkande företag och grossister och sånt där. Så det är komplett för den delen.

T: Vi upplever ju att verkligen lever på det vi producerar. Och...att..det som implementeras ute hos våra kunder görs också utav vår organisation, så Peter och jag tillhör ju då Intentias "fabrik", "research and development". Vår fabrik är väl närmare 350 personer ungefär...worldwide...som levererar till dom andra då som är cirka 3000, jag har inte dom färsk siffrorna, vi är väl ungefär runt....vi har ju fått rätta mun efter matsäck här nu....så det är väl lite....jag har inte dom aktuella siffrorna nu men....worldwide 3000 anställda, 3.5 mdr i omsättning, opererar i 40-talet länder. Det är affärssystem som historiskt och fortfarande är det som vi är bäst på.....och.....

M: Innebär det att ni har en separerad konsultverksamhet...? Konsultverksamheten beställer egentligen på nåt sätt av er, det dom behöver för att lösa problem hos kund?

T: Ja..vi har ju sådana leveranscykler där vi...så att säga...hur det tas fram vad det är vi skall utveckla.

P: Ja...vi utvecklar ju en standardprogramvara som...och vad den skall innehålla, det bestäms ju av marknaden så att säga...det plockas ju in från konsultorganisationen, från säljare och genom marknadsundersökningar, allt möjligt så plockas det in så att...det här är...vad man vill ha i systemet. Så gör vi det i standardsystemet. Sen installerar vi det standardsystemet och det passar ju inte hos alla kunder 100% och då får dom kunderna.... Då gör våra konsulter anpassningar för det då, tillägg eller vad det nu är för någonting som behöver göras.

T: Men en stor del av arbetet i ett kundprojekt kan ju bestå i att lära känna kundens processer, så att därigenom veta hur vi skall ställa in inställningarna i vårt standardsystem. Så det blir ju mer och mer hela tiden att... egenanpassningar ute hos kund skall inte behövas, det går ju mindre och mindre och andra sidan så får du veta mer om systemet och hur du skall ställa in det och lära känna kundens inriktning. För därför blir det ju lite olika applikationskonsulter inom vår organisation som kan vara t.ex specialiserade emot bilindustrin... "Food and

beverage”, ”fast moving consumer goods”...”fashion”, det är också lite sådana här saker som vi har blivit mer och mer inriktade emot, förutom det som Peter nämnde här då med tillverkande industri.

M: Det är alltså konsulterna som har specialiserat sig på den branschen och har en hyfsat god insyn i hur den typen av processer fungerar. Då jobbar man med att anpassa standardsystemet till det så att säga ”så här brukar det vara i den här branschen” så att man har någon slags grund att stå på.

P: Systemet täcker bättre alla deras behov då, inte [skrap]...

T: Sen har ju vi en kundbas då på cirka 3000 kunder och det är ju klart att det känns ju gott att ha det i botten i dom här tiderna jämfört med att inte ha haft den kundbasen... och det är ju klart då att dom ligger ju då på tidigare versioner av Movex då, vi har ju vart med då ända från system 36 på IBM, sedan på AS/400 och nu opererar vi ju på flera andra plattformar och burkar då. Och en stor del av våra kunder betalar licenser för att få framtida uppgraderingar när dom känner att: ”nu är det dags att vi måste hänga med rent internationaliseringsmässigt” eller utöka verksamheten eller operera på flera marknader o.s.v så kan ju dom då uppgradera, det blir ju projekt kring uppgraderingen då med tillgång till dom här nya licenserna då som ingår. Det är också en stor del som håller igång oss och driver vår utveckling vad vi skall göra, alltså vad kundernas val är, befintliga kunder också. Ehmmm.....

B: Nää, men det är bra för att som jag sa, vi får ju bara den bilden av Intenia som media visar eller som marknadsförarna vill att ni skall visa.

T: Det är klart man upplever ju den som.... Man kanske får den press man förtjänar men det har ju varit ganska tufft nu, jag vet inte hur ni upplever det när ni har gjort er info innan, vad ni har hittat för information om Intenia och så...

M: Alltså det är ju... egentligen, om man skall försöka ta ett helhetsgrepp på det så... det är ju ingen som klarar sig nu, alltså dom här åren som vi har gått på skolan och som vi har så att säga haft kontakt med företag och läst om branschen. Lien har ju gått hela vägen det spelar ju ingen roll vad man heter utan det är ju mörkt för alla.

P: Intenia har ju alltid varit ganska anonymt i pressen, går inte ut så mycket av någon anledning, ändå är vi ett av dom största bolagen då inom det området...

T: ... I världen då alltså...

P: Man kan tycka att vi borde synas mer och att folk borde känna till bättre vad vi gör.

T: Hela vårt segment då, vi är ju tvåa i europa och sexa eller femma worldwide, den[bilden] är ju ganska fragmentiserad så det är klart att SAP är ju störst, och sen kommer ingenting och sen kommer ingenting och sen börjar det tuffa till sig då med JD Edwards och dom här då, men jag tycker ändå det är rätt fantastiskt att IFS, Intenia och IBS här då i Sverige, att vi än en gång visar som så mycket annat att vi kan hålla oss så pass väl framme.

M: Ja det får man säga. Sen är det ju så också att finns det en sådan ledare i branschen som SAP så är det klart att dom får ju oproportionerligt mycket också då, det får dom ju alltså.

T: Mmm.... Jag tog bara en bild, [sätter igång projektor]. Jag skall visa en bild på just den här utvecklingen och det som driver den. Här är...ett litet sätt att åskådliggöra det....men det är ju alltid kul att utveckla, det finns ju alltid grejor att utveckla, både tekniskt och funktionellt och affärsdrivet då...men "end customer" för oss här det är våran konsultorganisation, våra business partner då, som opererar. Där har ni ju "business group" [pekar på skärmen]. Du har Northern Europe, du har Western Europe, vi har Asia vi har America etcetera. Dom levererar vi till och sedan är det deras sak att implementera, ha tillräcklig kunskap så att dom kan driva det som vi levererar. Dom skall hålla sig uppdaterade på teknologin och dom olika affärs....vad är det man behöver inom "automotive" i en framtid, hur mycket växer den marknaden? "Fashion" likaså. "Fashion" för Asien för oss gå ju otroligt bra, tydligen, och växer där då. Och där det liksom olika inriktningar inom "Fashion" också där man...ju mer man öppnar upp, ju mer specialiserat kan det alltid bli. Så dom får ju ge någon form av input då, nånting som vi kallar för "opportunity management". Där sitter olika produktchefer för både funktionellt ansvariga för kanske ekonomi, för funktioner, serviceorderfunktioner, orderflöde, material och produktionsstyrning och så vidare, och så vidare. I princip kan man säga att vi har nästan allt administrativt i systemet...det kan man säga...whatever. "Operations management", dom är en form av....det är dom som bestämmer över oss, mig och Peter, alltså resurserna in till vår fabrik, IRD [Intentia Research and Development], där man då får göra en utvärdering utav....har vi råd med detta? Är det långsiktigt, vad skall vi satsa på kortsiktigt och vad är affärspotentialen, helt enkelt.

M: Det innebär att dom filtrerar egentligen...?

T: ...vad våra produktägare ville.....d..d...d...det blir alltid självförsörjande....liksom är vi 600 skulle vi säkert haft också hur mycket som helst att göra. Hade vi varit 900 så likaså. Vi ligger lite...det blir en filtrering och en fokusering i detta steget. Sen körs det igång då ett antal projekt som vi driver här då. Ehh.. och som alltid kontinuerligt driver...sen har vi ju då...i våra tidigare versioner behöver vi ju då olika supportfunktioner för att olja maskineriet så att det fungerar. Det kan ju vara alltifrån inköp av datorer till att se till att vi har tillräckligt med projektledning som kan stötta det här...ehh...vi har legala krav ute i alla länder, jag menar EU-anpassningar, 10 nya länder som skall in här nu. Det gäller att vi agerar på det, för att ge nåt exempel då. Och så vidare....man skall se till att människor trivs då, "human resources". Men då går det in projekt, där vi kanske kör prepareringsprojekt...ehhh...tydliggör vilka utvecklingspunkter som skall tas fram, "define requirements", går in i en softvaruproduktion, samtidigt börjar kanske marknad titta på och följa med projektets syfte. "Vad är detta?" Hur skall vi så att säga saluföra och presentera detta ehhh...så att säga...i vilket spektra kan man sätta det? Och det här har man ju förstått att det blir nånting helt annat när dom sätter sin marknadsföring på det hela. Ehh...utbildar säljare och så vidare. Ehh..."product services" det är ju så att det är lite av det vi håller på med. Ehh...vi har en egen utvecklingsstudio som driver våran javautveckling och det är det frameworket bland annat då som skall se till att softvaruproduktionen, det man skall tillverka och leverera, kommer fram. Packning och skeppning av produkten, versionshantering...ehhh....kunna hålla ordning på hur vi har skickat dokumentation...installation, våra mottagare, och dom skall installera lösningen ute hos kund. Och samtidigt då när vi har sålt det, måste vi ha en organisation för "maintenance" och support här då som ligger kvar under produktens livslängd. Och så har vi garantiåtaganden omkring det och sen är det ju då tänkt att när vi har nått x antal tidsenheter så skall ju kunden vara triggad av nånting nytt som har kommit in här, och vara inne i den här loopen hela tiden. Såå...ja...det är ett sätt att visa på det då....som fabriken och dom här andra, vi 300 här och dom andra 2700, 2900 som håller på att jobba med utveckling. Så det är ganska stort.

M: Kan man se som att det som ni betecknar fabriken här då, att det är så att säga en projektorganisation där det sätts samman för en... ”nu skall vi göra den här saken”...och så gör man det och sen så sätter man samman ett nytt projekt: ”nu skall vi göra detta” medans det som är ”supporting process” är en del av en linjeorganisation, man befinner sig i en linjeorganisation och stödjer det aktuella projekt som just nu pågår?

P: Det är väl en blandning...vi försöker mer och mer driva allting i projektformen, allt funkar inte riktigt alltid. Allt blir inte alltid det, det blir lite blandat...men....målet är att allt skall vara det [projektform].

T: Projekten vi kör då...projekt som är utav likartad karaktär, där vi tror att vi kan ta synergieffekter utav varandra, organiseras i någonting som kallas kluster, som är en virtuell organisation egentligen, så att dom lever så länge projekten är. Sen kan det mycket väl bli ett nytt kluster. Så därigenom, och där, inom dessa klustrena då så har man ju då projektavstämningar minst en gång i månaden, hur man ligger till i förhållande till leverabler, problem, tester och så vidare. Så rapporterar man upp där och får man då så att säga resurskonflikter i detta så refereras det upp till ”Operations Management” igen, så får dom bestämma hur man skall prioritera det. Och hela det här arbetet styrs ju via...vi är ju ISO9000-certifierade då, så hela den...processerna reglerar ju en del av hur det är tänkt att vi skall jobba då.

P: Vi levererar med jämna eller ja...fasta tidpunkter...när det skall ut en ny version, så jobbar vi mot det där.

B: Om vi skall gå in lite på systemutvecklingsmetoder... En systemutvecklingsmetod kan ju vara väldigt mycket. Det kan ju vara processövergripande eller ner på minsta gemensamma nämnare: ”nu gör vi ett klassdiagram”, eller en objektmodell eller vad man nu vill kalla det för och det skall ingå i en kedja av aktiviteter som skall göras men om vi tar det några steg bakåt, vad är en metod för er egentligen, vad skulle ni säga att en systemutvecklingsmetod är för någonting?

P: Jag vet inte om man kan ge en definition på det så, det har ju genom åren har det ju.. det är ju teorier egentligen bara...och dom är mer eller mindre bra, mer eller mindre löser alla problem i världen, men dom fungerar ju aldrig, teorin är ju aldrig likadan som verkligheten och det har det aldrig varit någonstans på dom ställen som jag jobbat på och jag har hållit på med systemutveckling sedan 1979 och på Intenia sedan 1990 och hållit på med allt möjligt, JSP[Jackson Structured Programming] och olika metoder, det är ju sådant som kommer och går. Vi kan inte säga att vi har en metod. I slutändan blir det en samling av lite bitar från här och var så att säga, som passar för det programspråk, den miljö man håller på och utvecklar i för stunden. Vi har ju.. nu är det Java men innan var det ju RPG bara och det ser annorlunda ut och då fick också metoden som vi höll på med vara lite annorlunda. Sen så styrs ju vi ganska hårt av den här ISO9000, där har vi ju satt upp att: ”så här är utvecklingsprocessen, det skall vara dom här stegen och dom här grejerna” och det måste vi ju följa då. Sen drivs det också ganska hårt av att vi har ganska standardiserat utseende på dom programmen, hur det skall hänga ihop hur det skall se ut, även programmeringsmässigt då.

B: Jag får lite uppfattningen av, när du pratar, att det är väldigt mycket kunskap som sitter i väggarna att det kan vara lite olika på ... att Kalle gör så och att Lisa gör så. I slutändan blir det en uniform produkt men...

P: Njaa, alla är ju fria att göra olika eller som man känner då så länge man följer processerna och man följer den standarden som vi har satt och den utvecklingsprocessen med dom steg och har dom här designstegen, detaljerad och övergripande design och alla de här grejerna. Man har testplaner och alla dom här grejerna så är man fri att jobba enligt egna metoder, vi har inte något speciellt krav på att man skall ha någon speciell UML eller sådant här... sådant har vi inte utan där är man fri att göra lite som man vill, att göra som man känner att man behöver eller som teamet känner att dom behöver. Det varierar ju lite beroende på vad man håller på med, det är ju liksom skillnad om man håller på med just affärssystemets grund...basdelen, eller om man håller på med webdelar eller något sådant. Då jobbar man lite olika men, summa summarum, så skall man ändå följa samma mall och därför är ju inte mallen detaljerad ner in på minsta bit.

T: Det är ju som sagt så att om du går in i ett affärsobjekt[Javaklass] i vår kod så är det ju klart att någon som jobbar på ekonomisidan känner igen sig även om han skulle gå in i ett affärsobjekt på kundordermodulsidan.

P: Jo, men det är...i och med att det är standardiserat, vi har standardiserat det så som det skall se ut sa jag, det skall innehålla dom här delarna.

B: Men hur ditt tankearbete hardet är en annan sak. Det är inte något som sparas i en pärm eller så utan det...

P: Nej, det kan man göra om man vill men det är inget krav från vår sida, men om man nu gör det och skriver det så är det ju bra om det står här. Och då har man ju möjlighet att lägga med nån liten notis, men det ligger liksom inte i kravet då.

B: Men då är det...ööh...jag tänkte inte ta fram den riktigt ännu, men nu blir jag ju tvungen! Vi hade en liten frågestund med oss själva...då ritade vi en liten modell på hur vi ser på det här med metoder, processer och så vidare. Och då...valde vi att kalla det övergripande för en process [pekar på vår käcka modell]. Om man tänker sig till exempel vattenfallsmodellen, den har ni säkert hört talas om...du har ett antal olika faser, du har analys, du har design, du har testning, implementering och i vissa fall avveckling. Det trodde vi var nånting som styrdes ifrån strategisk nivå, alltså nu säger vi, som ni sa...ISO9000, den kan jag inte men, jag kan tänka mig att den kommer så att säga uppifrån...planeringens sida..."nu kör vi såhär..."

P: Ja...det kan man väl säga. Man har bestämt sig, men det har ju liksom...själva processen där som har tagits fram, den har tagits fram med utgångspunkt hur vi jobbar och utvecklar va...det är ju en blandning av att den har kommit uppifrån och att det liksom har kommit underifrån. Det skall praktiskt återspegla det sätt som vi jobbar på. Men det är inte nåt...det...det är ju en kvalitetsstämpel på själva processen så att säga...inte...

M: Ja...den hanterar ju spårbarhet och sånt också.

P: Den...den skall ju då...den skall ju tala om...den skall beskriva hur vi jobbar. Och sen skall vi följa den och sen skall vi utveckla den under tiden, för det ligger ju också i den. Det är inte så att man har bestämt nåt uppifrån och sedan trycker på det, utan det är...man har ju bestämt att man skall köra det här och sen har man jobbat med att ta fram dom här

processerna, och då har man ju hittat hur vi jobbar verkligen och försökt att beskriva dom i dokument, ord och flöden och sånt där. Men den styr då.....

B: Sen så tänkte vi att inom den här processen så....finns det någonting som vi namngav som metodologi, som i sin tur kan innehålla olika faser.

M: Vi sa egentligen så här då att...om man nu tar till exempel en sån historia som vattenfallsmodellen, så säger den egentligen vilka faser man skall dela upp utvecklingsprojektet i till exempel. Men den säger inte hur man skall göra det. En metodologi däremot, kan säga hur man skall göra saker och ting. "I analysfasen skall vi göra det här och det här och vi gör det på detta sättet, med hjälp av vad det nu kan vara....use case scenarios...bla, bla, bla". En metodologi kan delas in i aktiviteter, där man säger såhär: "ja..nu skall vi göra use-case scenarios. Hur gör vi då? Jo..vi sätter oss ned tillsammans med våra kunder och går igenom hur de skulle utföra en viss del i det här systemet och utifrån det så kan vi rita nånting...". Då sa vi såhär att, den taktiska nivån då...det är ofta där som...från den nivån som man talar om metodologi, om vi var överens om att den strategiska nivån var den nivån där man bestämmer i vilken ordning man gör saker och ting, det kan gälla alltifrån projekt, hur ett projekt skall gå till. På den taktiska nivån bestämmer man sig för: "nu när man skall göra dom här sakerna som vi har blivit ålagda som ett projekt, vad skall vi använda för metodologi för att göra dom här sakerna, hur skall vi komma dit? Jo...vi använder oss av..."... vad det nu kan vara. Och sen då, när det gäller dom enskilda aktiviteterna...så på en operativ nivå: "Jo...vi skall göra ett use-case, men hur gör vi det?" Då ligger det mera ofta på det enskilda teamet, den enskilda utvecklaren kanske till och med. Det blir på något sätt olika granularitetsnivåer här, väldigt grovt och svepande, till exempel ISO9000, den ger oss strategiska riktlinjer för hur vi skall göra. Inom den så har vi ett vedertaget arbetssätt på vårt företag som säger att: "nu skall vi utföra dom här sakerna", och inom den då så finns det..."när vi gör den här aktiviteten så gör vi på det här sättet" antingen för att vår metodologi säger åt oss att göra det eller för att vi har kommit fram till att detta är det bästa sättet att lösa det på. Så såg vi på hela den här bunten kan man säga för ibland så... om man pratar om metodologier i allmänhet så finns det ju ett par som är av just den här typen, alltså dom säger egentligen inte mer än vad som skall göras med andra är betydligt mer explicita och det finns t.o.m dom som är väldigt explicita och säger att: "om ni inte gör alla dom här aktiviteterna på det här sättet så får ni inte det här slutresultatet och då blir det inte bra". Så såg vi på det och vad vi ville göra med den här var att använda den som någon typ av stödmodell för när vi pratar om vad är en metodologi och vilken nivå rör vi oss på här...

B: Och det jag ville komma fram till var just att... och rätta mig om jag har fel nu... att den operativa nivån, den är ganska så fri, varje utvecklare är ganska så fri... precis som du sa förut, att så länge ni når målet, vad det nu må vara, så är ni fria att använda vilken metodologi ni vill eller ingen alls eller er käcka modell som ni själva har kommit på. Huvudsaken är att ni strukturerar er verksamhet på ett sätt så att ni kommer fram till målet.

P: Så är det mycket.

T: Ja, det stämmer.

M: Ja, det är bra för nästa fråga är ju ganska intressant i det läget då, eftersom vi pratade om det här med metodologi förut så ... vad är för er en god metod? Vad skall en metod innehålla för att den skall vara bra, eller vad är det som gör att en metod inte är bra?

P: Ja det är ju svårt att säga, det är ju egentligen bara att det fungerar i verkligheten, det är ju liksom att... det skall vara smidigt att jobba enligt den och man skall kunna få hjälpa av den. Man skall inte belastas, inom citationstecken onödigt arbete. Eller åtminstone, det onödiga arbetet skall då vara beskrivet så att man förstår nyttan med det så att någon annan kan dra nytta av det om man själv inte kan dra nytta av det. Det är väl... att det blir ett fungerande bra slutresultat av det. Det är ju egentligen det som är en bra metod.

B: Hur ser vi på... Om man jobbar enligt en viss metod och har jobbat enligt den många gånger så vet man ju så att säga vad det är som ligger framför en och man kan göra en uppskattning på hur pass lång tid.... Ja, vart i tiden ligger vi rent projektmässigt. Det är också en sak som jag kan tänka mig är viktig.

P: Jo, det är det ju alltid.

B: Men om man då är ganska fri på den operativa nivån att så att säga gå efter sin egen metod, hur styrs det då på den lite övergripande nivån för att... ni arbetar ju under tidspress, ni måste ju så att säga ha koll på, vart befinner vi oss i processen.

P: Jo, men det görs ju via regelbundna återrapporteringar eller...ja, man talar egentligen om hur lång tid man tror att man har kvar. Då justeras det ju och sen får man ju se om man klarar det eller hur det ligger till. Sen kan det ju ske omprioriteringar eller ytterligare resurser eller man kan bestämma att: "nej vi håller detta och väntar med detta för vi kommer ändå inte hinna inom ramen för när det skall vara färdigt, vi får skjuta på det" eller vad som helst.

T: Ibland kan det ju bli avvägningar mellan...så att säga minska på det ursprungliga scopet. Eller egentligen... eller tillföra mera resurser eller förlänga projektiden. Som Peter nämnde här förut, så jobbar vi väldigt hårt efter vissa editionsplaner som vi skall gå ut efter, två släpp om året och vissa saker skall bara vara där på plats. Sedan följer då utbildning av kunder, presumtiva kunder och så vidare här då.

M: Kan man se dom här två punkterna om året som punkter kommande från den strategiska nivån?

T: Ja, definitivt.

P: Ja..det kan man göra.

M: Man rättar sig efter det och man anpassar sitt arbete efter det...?

P: Ja.

T: Ja.

T: Sen då exempelvis när saker och ting skall fogas samman så har man ju vissa låsningsdatum, dom är heliga. Där sammanförs då så att säga dom olika affärskomponenterna. Först ligger man ju i en utvecklingsmiljö, sedan, med jämna mellanrum så flyttas dom över till en verifieringsmiljö, regelbundet. Sedan sker det ju verifiering och acceptanstester på där då, på den miljön som kommer att bli den skeppade miljön. Där har man liksom en utvecklingsiteration också inom oss.

M: Och kunden är med...i acceptansfasen, naturligtvis?

T: Nej, inte just där, men...sedan då så kommer ju sådana här alfa-, beta- och gammatester. Och vi...efter...som man haft dom här verifikationstesterna internt så utser man på vissa produkter då...ehh...pilotkunder som ehh...valts ut då. Det kan ju vara produkter som...är i framkant och dom mycket väl kan tänka sig att bli först och då kan man ju sätta in mer resurser då så kanske vi har chans att påverka en del av utseendet på det. Såå...utses det ju då vissa produkt...pilotkunder då...så kör man ett varv med dom och sedan så blir det så att säga en "general availability" på det, där det släpps då för hela konsultorganisationen och då går dom på...vad skall man säga...regionala utbildningar då där vi har ett, som vi kallar det, Intenia University på kurser när det är "general availability", för då anser vi att vi har en sådan stabilitet i det att det bara är att tuta och köra då. Men det är ju en hel del, just det här med aktiviteterna, du nämnde ju det här med tester och sådana här saker...där kan man säga att...diskuterat inom...hur man kan automatisera och benchmarka och få liksom till det här. Det är ju resurskrävande att säkra upp allt detta.

P: Men det är ju en del i processen så att säga, att det skall vara dom här testerna. Och sen så är det ju lite olika steg. Vi har ju en blandning av vattenfallsprincip och mer iterativ så att säga...den går ju vissa steg, men innehåller samtidigt en tillbakakoppling så att säga, där man ser sdjhfdshhsdf[otydligt] man behöver. Det är en blandning av dom.

M: Om man skall återgå till bilden så kan man säga att det är en blandning av vattenfallsmodellen som finns här[pekar] och där iterationerna kanske befinner sig här[pekar]. Så inom varje steg i vattenfallsmodellen kan ni då göra fler iterationer...tills ni tycker att ni är klara och då gå till nästa steg, och då kan nästa steg innehålla fler iterationer....

P: Och den kan då, om den visar sig först, så kan det gå tillbaka till första och vi kan börja om där, men...men...

M: Men det är fortfarande så att ni följer de övergripande...till slut...så hamnar man där. Sen i vilken ordning man gör...när det är klart...det är dom här stegen som skall göras och sen kan det se annorlunda ut därinne, då.

P: Jo...

M: Om vi skall tala litegrann då om det här...vi har identifierat, när vi har suttit och diskuterat det här på våran egen kammare....så kan man ju se det som att det finns ett gäng olika typer av metoder:

Det finns dom här metoderna som man går ut och köper, till exempel RUP.

Man köper ett paket, och i det finns det verktyg för att göra vissa saker.

Dom metoderna kallar man för out-of-the-box-metoder, dom tar man och använder dom och så blir allting bra...

Sen finns det metoder som är mera dynamiska i sin sammansättning. Det finns ett sätt att lösa det här på som kallas "method engineering", där man har en eller några människor på ett företag som kan en hel del om metoder och är konversanta i olika metoder. Dom bryter man ner i vad man kallar "method chunks", småbitar, självständiga småbitar, som man lägger i sin verktygslåda. Så ser man att: "okej, nu skall vi göra ett utvecklingsprojekt här. Vilka av dom här method-chunksen vill vi ha med för att lösa det här projektet?" Så plockar man små bitar och sätter ihop dom så att dom fungerar för det här projektet som innehåller det som man behöver, sen är det bra. En annan variant är någonting som kallas för DSDM som är ett större

ramverk egentligen, det ingår i någonting som brukar kallas för contingency-teorin och det är lite grann åt "method engineering"-hållet där man har ett gäng hela metoder och väldigt många på företaget eller organisationen kan alla metoderna och så väljer man blanda dessa metoderna för varje inkommande projekt, man delar inte upp dom utan man tar antingen en metod eller en annan. Det är dom grova typerna som finns då... och uppenbart från den här diskussionen är att ni har ju i alla fall inte någon out-of-the-box metod, men vad som är intressant är kanske hur har den metoden som ni själva använder er av, om den nu heter någonting eller om ni inte kallar den för någonting utan bara är ett arbetssätt, hur har den växt fram?

P: Den heter ju ingenting, det är ju en sorts ihopplock av metoder eller delar av metoder som man har funnit under årens gång att dom fungerar och är bra och fungerar för oss. Ofta så bedriver vi ju då när det är helt nya grejer, och man kanske testat lite sådana här saker så kör vi prototypprojekt då med mindre grejer och då kanske dom använder sig av några sådana här metoder som man har plockat upp någonstans ifrån då och då testat man av dom och sedan tar vi då det som vi tycker passar och är bra, då kan vi ta med oss det.

M: Är det en formell process?

P: Nej, den är ganska informell.

M: Vi talar egentligen om ett organiskt framväxt arbetssätt?

P: Ja, det är ju att någon har hört talas om eller känner till eller sett eller "det här kan man använda", det här är bra att använda här och så provar man det då.

T: Det skall ju fortfarande åskådliggöras via... det skall ju inte vara en sådan som vi kallar för gerilla-utveckling... Det försöks ju i och för sig alltid, för man kan ju brinna för någonting och agera utifrån vad man tycker är rätt och riktigt och driva utvecklingen, men det måste ju ändå upp och synliggöras och drivas i en projektform... det blir ett prototypprojekt då, då får man se det som en resursaktivitet.

P: Det har passerat många sådana där grejer under årens lopp eller verktyg som Rational Rose, Visual Age, alla möjliga, JSP, sådana där grejer. Det har plockats från dom delarna som har fungerat för oss, det är liksom inget som man har tagit rätt av, som man följer till punkt och pricka.

B: Men om Mattias här skulle börja jobba, inte för att det skulle hända nu när marknaden ser ut som den gör men...

[skratt, alla]

Vi ponerar detta, och så kommer han som helt ny in i huset. Hur har får han.... Hur lär han sig hur han skall dokumentera, vilka leverabler det är som skall... finns det samlat i någon pärm... Jag är lite ute efter... den här kunskapen, vart finns den?

T: När det var... du får hjälpa mig här Peter..när det var som värst eller bäst, när vi växte utan att köpa företag eller så... den ena året växte vi med 30% det andra med 40%. Det var ju positiva problem men det var väldigt mycket positiva problem, det var ju stora traineekullar som kom in som gick traineeprogram på 9 månader, oaktat egentligen om vi så att säga...om man skulle in på IRD[Intentia Research and Development], vårt utvecklingsbolag, som vi är på eller någon av våra global partners och då fick ju dom gå igenom olika steg. Bland annat:

hur ser vår programmeringsstandard ut, hur dokumenterar du den, var finns dokumenten om det. Det var en del. En annan del var ju: vad syftar våra affärsprocesser till, hur gör vi våra kunder nöjda, hur gör vi dom glada, industrikunskap. Liksom i den rollen som dom skulle ha och sedan då naturligtvis att lära sig att kunna köra Movex[affärssystemet], att kunna ställa in systemet, kunna köra flödena, kunna demonstrera det och utbilda. Och...det är klart att skulle man in på IRD så blir ju det tekniska fundamentet betydligt större då. Det var ju en del i det hela. Alltså formell utbildning, där det var mycket korvstoppling och sedan kombinerat med mentorskap och liknande, då.

M: När man kom, så att säga fram då....

T: Ja...precis. Så det...tycker jag att vi fick...och har...eeehhh...har bra kontroll på, för att kunna växa då, om det skulle behövas igen. Via Intenia University då. Sen har man ju då olika roller då...jag är ju projektledare och Peter är ju utvecklare, vi har designers och systemutveckling, det finns ju affärskonsulter som är kitt mellan projektledare och utvecklare, som ställer in Movex, kör det, men som inte kan felsöka under motorhuven. De kan läsa av och konstatera att oljan är slut ungefär...men sen får man ju jobba lite teammässigt här då och...och motsvarande inom IRD också, hur vi jobbar med olika roller då. Så rollerna är en viktig del i då...för att kunna ta emot rätt kunskap, men....

B: Det jag var ute efter lite...var just det att om man kommer utifrån, är nyutexaminerad eller kommer från ett företag eller vart man nu kommer...kommer ny in i organisationen, om man kommer till en organisation som kör en metod av typen out-of-the-box, säg att ni kör RUP med UML, så är det ju väldigt lätt för nån att knappa in amazon.com, beställa hem några böcker och sätta sig och läsa: "så är dokumenteringsstandarden". Sen kanske man inte förstår metoden för det, för det tar en viss tid att få förståelse för det, men man kan åtminstone se att: "det här är slutprodukten". Medans om man kör en egenutvecklad metod så kan det vara ett problem, men det upplever ni inte att det har varit, tack vare den här trainee.....

T: Man kan ju alltid bli bättre. Inom vår utveckling här då så...vi skiljer...vår organisation är i alla fall historisk, fast det börjar ändras lite för vi har en utvecklingsenhet som sysslar med utveckling av affärskomponenterna och sen kan man säga att underliggande lager i Foundation och databaslager, VisTooling som behövs för affärsutvecklingen...den har haft kanske lite friare spelrum. Det har varit hårdare reglerat inom development, alltså dom som utvecklar affärsapplikationen, själva Movex, kärnan. Ja...det blir lite luddiga svar hela tiden....det är inget svart och vitt.

M: Så är det nog. Det är sällan svart och vitt och det skall nog inte vara det heller.

B: Den typen av frågor vi har här, det är diskussionsfrågor. Det är därför vi inte har mailat ut dom heller för att det finns inget rätt eller fel svar. Det går inte att förbereda sig på det. Det är hur ni ser på verkligheten.

M: Om vi då nu skall försöka, det kan ju vara roligt i och för sig, att ta ett rent praktiskt exempel. Man kan tänka sig såhär: från början var Movex inte utvecklat i Java, utan det låg andra programmeringsverktyg/språk i botten. Då är det...när ni bytte språk till Java, fick det återverkningar på hur ni utvecklade...alltså vad ni använde för, kalla det metod, hur ni gjorde? Överförde ni bara det gamla arbetssättet på det nya och det funkade precis lika bra?

P: Naej...ja..det ändrade det ju sig, för det var ju inte likadant. Samtidigt försöker man ju ändå behålla så mycket som möjligt, men i stort är det inte så mycket...jag menar...man ser dom stora processerna, stora metoderna, men mycket är tankearbete för att införa en sån här stor grej. Där skiljer det ju inte så mycket utan det är ju sen när man kommr ner till detaljer egentligen, som det ser annorlunda ut.

M: Den anpassningen som ändå skedde då på detaljnivå, hur så att säga kom man fram till att man skulle göra på annorlunda sätt på den nivån?

P: Det gjorde man ju i och med att man började... det här var ju en process som tog en bra tid och under tiden där då så drev man vad skall man säga team, som jobbade och började titta på det här då, att komma över Java och jobba enligt det. Komma in och känna på det och det var ju egentligen då som man mejslade ut hur man skall jobba sen och efter det har man då utbildat resten av den stora kadern programmerare/utvecklare då på dom nya metoderna, det nya sättet och under den tidens gång då så bytte vi ju eller utvecklade nya verktyg som passade för nya Javamiljön. Utifrån dom som passade i den gamla miljön och det är ju också...

M: Och det var vad skall man kalla det ett evolutionärt utvecklande?

P: Ja, det kan man säga.

B: När vi nu kommer in på skruv och mutternivå här så blir programmeraren inom mig plötsligt intresserad. Vad kör ni för utvecklingsmiljö?

P: Vi kör en egen utvecklingsmiljö. Som innehåller dom grejer som behövs... det är ganska standardiserat, vi jobbar med språk till exempel, det skall fungera för alla världens språk och så vidare, det finns krav på kontanter, meddelanden allt sådant där, och det stödjer den... hyfsat lätt att översätta och så där. Tabeller, det är också standardiserat. Det skall vara så här och vi har verktyg som tar fram det utseendet.

B: Det har tagit ett tag att ta fram den utvecklingsmiljön...

P: Ja... men den ändrar sig och den lever hela tiden och sen är det ju nya delar så.... Vi har... vår visuella del, verktyg för det.. standardiserat på ett visst sätt. Ett visuellt lager som har ett visst utseende och det ställer vissa krav på att man skall göra på ett visst sätt och då har vi verktyg som stödjer detta och man kan inte göra något annorlunda... Likadant med utskrifter och sådant där... när man kommer till programdelen, själva Javanivån, då är det ju en editor vi kör och där kan man köra med lite olika, men den är inkopplad i vår miljö.

B: Tror du att det kan vara.... Nu kom vi ju in på ett sidospår som faktiskt kan vara någonting vad gäller metदानpassning. Tror du att ni har en så pass egen systemutvecklingsmiljö som ni faktiskt har och att den är... som du sa... skräddarsydd för att det skall gå att göra på ett visst antal olika sätt... tror du att det är därför ni klarar av att ha en väldigt fri styrning på den operativa nivån... om utvecklingsmiljön inte tillåter att man kan göra på något annat sätt vad gäller utskrifter som du sa och en grafisk design så blir det ju att man gör på det sättet. Det går inte att göra på något annat sätt.

P: Ja det hjälper ju till och det är ju liksom en förutsättning... det ligger liksom i alltihopa med den här standardiseringen, det gör ju också att vi kan hyfsat lätt förändra saker som

gäller.... Så att säga massförändra dom, att vi vet att det ser ut på ett visst sätt och då kan man lita på det och göra sina förändringar och bygga ut och ändra. Vi har ju.... en "foundation", en grund, en Javagrund i botten som stödjer resten av det hela då.. den kan ju ändra sig då, det kan hända att man vill göra något... att den skall se annorlunda ut eller så. Tack vare att det är hårt standardiserat har man ju möjlighet att göra det. Även i våra verktyg så bygger vi verktyg som kan göra detta, som i viss mån kontrollerar att det ser ut som det skall göra. Så visst...det är ju en stor del i det hela, det är ju liksom en form av inbakat i metoden så att säga...allt är ihopbyggt.

M: Det är nästan så man skulle kunna likställa utvecklingsmiljön med metoden....?

P: Det ligger en viss form av kunskap och....inbyggd i det då så att säga....som då inte belastar....som man bara har att rätta sig efter.

M: Den är implicit på det sättet att den inte syns utåt på något sätt.

T: Men den går ju in där.....den kan ju då....regelverket är ju metadatastyrt....så kan ju den då vara olika för olika versioner, fastän vi kör på samma version av utvecklingsmiljön...och via arvet i Javan och såna här saker. Peter nämnde språken, men även för att kunna hantera dom här lagliga kraven då i dom här olika länderna som är en oerhört stor grej i till exempel ekonomisystemet där man har 100 olika momsatsar i Italien, dom är ju totalt fria i sitt tänkande , men när det gäller saker så är det en jävla koll på det, va....hantera språk i Israel och såna här saker, det är...då blir det mycket att hantera och hålla ordning på vad man har släppt igenom.

B: Å andra sidan, vad gäller Israel och västbanken och så, så är väl språket det lättaste att hantera eller???

[skratt, alla]

T: Jag är lika förvånad över att dom har tid att sköta vanliga jobb, kan slappna av...

M: Om det vore så...att man i ett team, i ett enskilt projekt, skulle utveckla en del i ett system, en modul eller vad ni vill, och man stötte på något man inte stött på förut. Och då insåg man i teamet att: "Mjaaee...vi kan inte göra såhär som vi är vana vid att göra." Vad gör man då?

P: Jaaa...det beror sig på vad det är för typ av projekt, så att säga. Är det till exempel ett prototypprojekt, då kan man väl göra som man vill, det är väl inget mer med det. Är det ett vanligt utvecklingsprojekt och....då får man man föra det uppåt egentligen...då skall det egentligen gå hela vägen upp till att begära ändring av våra processer, då...om man nu tvunget måste göra på ett nytt sätt. Så får man göra så sen då, men då får man ta med det, och så kommer det med i processen, vårt sätt att arbeta då så...och då är det ett sätt som gäller sen då...

M: Då är det formaliserat på det sättet....

P: Mmm...ja....

M: Om det nu är så att man har ett gäng, samma gäng, utvecklingsteam, som får en uppgift som dom skall lösa. Dom sätter sig ner och funderar på hur dom skall lösa detta så har dom

antagligen med sig någon typ av verktygslåda. Den här verktygslådan, är den...alltså...antag att man hade ett team som har jobbat med detta sedan -79, som du sa...varit anställda på Intenia sedan -90, så är det förmodligen så att det teamet hade haft en betydligt större verktygslåda än det teamet som hade varit anställda sen -97. Finns det någon...företagsgemensam verktygslåda som man använder sig av eller....får dom använda dom verktygen dom har med sig av erfarenhet eller....finns det någon formaliserad verktygslåda som...dom har möjlighet att...."dom här metoderna finns ju också här, så dom kan vi använda oss av"...går det på erfarenhetsbasis egentligen?

P: Det går nog mer på erfarenhetsmässig basis, vi har ingen sådan katalog direkt där man slår upp vad man kan använda nu för detta, utan det är erfarenhetsmässigt... ofta kommer man ju inte in... det är ju inte ett team av bara nya så att säga... utan ofta kommer man ju in i något gammalt team då går man till dom och frågar eller man hör sig för...

T: Varje projekt knyter an till dom här olika rollerna då som har... för att stödja systemutvecklingen... i varje projekt så dels har du en kund, så även om kunden är intern så är det ju ändå kund som det skall levereras till och uppfylla dom kraven som finns, sen i projektet så har man ju projektledare och utvecklare men dessutom då så här med erfarenhetsbiten, så har vi ju en projektarkitekt som är alltså ytterst ansvarig för... kalla det då... kan man inte göra si eller så... men då är det projektarkitekten som har sista ordet, i dom flesta fall så är det inget problem utan det blir konsensus, utan det är ju ändå formaliserat att projektarkitekten har det yttersta ansvaret för att ta vissa beslut då för hur det skall göras.

B: Han är en form av teknisk projektledare eller?

T: Ja, han skall besitta tekniskt kunskap som är väldigt djuplodande.

M: Om man går tillbaka lite till den här teamdiskussionen som vi hade här förut... sker det intrakommunikation mellan dom olika teamen?

P: Ja det gör det och den är ju givetvis bättre ju längre man varit med, men jo, det gör det ju. Sen har man ju lite... även formella möten mellan teamen och sen sätts ju...för olika projekt sätts det ju ihop folk från olika team och då blir det ju också att man får ett naturligt utbyte, man får input från olika håll och man vet vem man skall prata med och...

T: Men vi har ju även byggt databasstöd för detta då, dels inom vår egen organisation inom IRD där du skall kunna söka på specifik kompetens men speciellt då när du skall belägga projekt ute i konsultorganisationen så har du ju alltså stöd då du kan gå in i databas och säga så här att: "jag behöver en programerare som kan C++ med tre års erfarenhet som finns i Asien, har jag chans till det mellan mars och juni?" För någonstans när du ju en kritiskt gräns, där det här informella måste få ett tekniskt stöd och det är framför allt mycket sådana grejer med resursbeläggning så att vi har och gör hela tiden och utnyttjar rätt resurser för att sätta ihop rätt team. Sådant stöd finns ju också då.

B: Om man tittar lite på vilka leverabler som levereras uppåt i hierarkin så... givetvis skall den färdiga produkten, koden eller den del som skall vara färdig för att just den här testcykeln skall göras, det skall ju den operativa nivån lämna ifrån sig men vad... om vi ser i rena papper, vad är det som lämnas ifrån den operativa nivån upp till den taktiska, eller du som projektledare då, för att du skall kunna ha ett grepp om situationen.. är det några dokument som lämnas till dig för att du skall kunna...

T: När ett projekt är avslutat så kan man ju säga.. eller när ett projekt påbörjas så definierar man också vilka leverabler som skall komma ut ur det projektet och det kan lika gärna vara ett leverabel som säger att man har iordningställt en utvecklingsplan, en utbildning. I vårt fall, när vi levererar en ny version av våran utvecklingsmiljö så har vi uppdaterat dokumentationen, vi har packat vår nya version och testat den och så att säga levererat den och satt den i produktion. Då är det en leverabel. En annan leverabel kan vara att vi skall iscensätta en utbildning och genomföra den på utvecklarna så dom vet vad det är som har förändrats. Det blir ju både mjukt och hårt. Så det är ju både kod, dokumentation, utbildning...ja det är väl dom som återkommer egentligen...och så i denna processen skriver vi ju även projektavslutningsdokument där man tickar av att det stämde överens med dom leverablerna som man kom överens om i början här.

P: Det är väl också nånting som styrs av ISO9000-processen....som säger: "Det här är processen och metoden", den innehåller ju lite olika steg. Sen står det i den att när man startar upp ett projekt så talar man om vad det skall leverera ut och sen kan det ju vara en prototyp och inte mer med det, eller det kan vara dokumentation eller en blandning. Eller det kan vara bara ett papper som går som beslutsunderlag, ja...beroende på vad det är för projekt.

B: Är det vanligt att den systemtekniska dokumentationen, alltså nu tänker jag inte..."fyll i dom här rutorna så får du det här svaret, och så får du en prognos över 2004" eller vad det nu kan vara, utan jag tänker snarare saker som användarfall, objektmodeller och så vidare. Är det...följer dom med och in i en pärm och vidare....eller....är det nånting som dör....?

P: Den lagras ju då...vi har ju.....alla projekt går ju...får ju ett hem i maskinen så att säga...man skapar ett projekt med en mapp, och där finns alla dokument. Allt samlas ju där. Allt kan du lägga där, och sen hänger ju det ihop och finns ju kvar. Så det...även såna grejor som modeller, om man har gjort några såna, samlas där.

B: Men det är inget som uppdateras sedan ni har levererat produkten, om det skulle vara så att det sker en anpassning.....?

P: Njae....då kan man ju hämta därifrån, från det förra projektet då och ta med det in i det nya...ja...så lever det där då, så att säga.

M: Det reflekteras inte när ni har släppt iväg den, som en färdig vad det nu är...så, om det är så att applikationskonsulterna sedan gör ändringar, som egentligen skulle....det uppdateras inte hos er....den är död när den är levererad.....?

P: Nej...den lever sitt eget liv, sedan då va[den ändrade dokumentationen].

M: Hos dom[applikationskonsulterna] då, så att säga...

P: Jaa...där har dom....där sätter dom...där har dom en process där dom för in sina anpassningar eller implementationer, har dom en modell och en process för. Där skall dom då hålla sådana grejor då, såna ändringar som dom har gjort eller lagt till eller.....

M: Ja...

P: Sen kan ju det komma till nytta då, om vi bestämmer att det här var en så bra grej, och alla eller många, väldigt många använder det, så det skall vi ta med i standard. Då kan ju dom grejorna komma tillbaka och komma in i utvecklingen.

M: Men då är det deras grejor som är så att säga input till det projektet, som blir när ni skall inkorporera det i eran standard...?

P: Ja...

M: Vi har varit inne på det litegrann... när vi pratade om teknikskiftet från något gammalt till Java-biten så att säga... vi sa att faktorer från den strategiska nivån som driver anpassning är sådana saker som omvärldsfaktorer, hur ser marknaden ut, vad gör våra konkurrenter det kan till och med vara sådana saker som.. och det skall man inte underskatta även om det låter fjantigt... sådana saker som trender, alltså nu skall det vara så här, nu är det RUP som gäller.

P: Jo....

M: På den taktiska nivån kan det vara sådana saker som intern organisation, om man gör en stor omorganisation i företaget, av någon anledning, man divisionaliserar sig eller så... Så kan det också vara anledningar till att man väljer att anpassa sitt arbetssätt, sin process och sina metoder. På den operativa nivån så kan det vara sådana saker... där är det lite mera luddigt då.. men där är det sådana saker som medarbetarnas kompetens som driver anpassningen. Om man har medarbetare som kan väldigt mycket om väldigt många olika saker så är det kanske ett arbetssätt man väljer, har man inte det så får man kanske styra upp det mycket hårdare och välja ett annat sätt. Hur upplever ni det? Finns det drivande faktorer som ni ser rätt av att: "ja så är det?"

P: Ja, det finns ju egentligen på alla nivåer...

T: Ja, jag känner igen mig klockrent, som exempel betraktat så stämmer ju det på varenda nivå.

M: Jasså, det gör det?

T: Ja, det gör det.

M: Och då kan vi ta ett exempel som vi egentligen redan har tagit. På strategisk nivå är ju just det här med ISO9000, vi bestämmer oss för att vi skall certifiera oss och det får återverkningar för arbetssättet. En annan sådan här grej som är intressant som vi har stött på bland annat på...

T: Ett annat exempel som vi tittar på här uppe nu då, det är ju det att man via... om man behöver en modifiering eller hur man skall sätta våra system så har ju vi... processmodeller som beskriver hur det håller ihop och där kan man ju säga att vi håller på bland annat, vi gör lite researchaktiviteter och tittar på att grafiskt representera hur dom olika affärskomponenterna håller ihop och där igenom via ett grafiskt gränssnitt, utan att man sitter i en dialog med kunden, kan sätta systemet och alla parametrar och hur det skall köras då med automatik. Det är ju ett sätt att så att säga...att det är drivet, att det är många som tittar på det och att man hör att många håller på och tittar på det och det kan ju bli lite stress att alla håller på och skall springa åt samma håll då, även om det inte är det bästa.

M: Ja det kan ju bli så...jo vi hade ju pratat litegrann med ADB-kontoret och dom har för övrigt valt att använda sig av RUP numera som utvecklingsmetod, och dom gjorde det i samband med att man gjorde en stor omorganisation. Ibland tenderar det att vara så att när man ändå så att säga gör en stor omorganisation, och det är mycket kaos innan det sätter sig igen, då finns det vissa drivande krafter som säger att: "skall vi inte passa på att göra lite uppdateringar och omändringar" eller "skall vi inte ta tag i det här med metodanvändningen och se till att det blir likadant i alla delarna i företaget?". Har ni varit med om någon sådan utveckling eller märkt av just det här med att organisation eller sättet hur man organiserar sig driver valet av metoder?

P: Ja, det gör det ju och det har det väl gjort... vid några tillfällen då har det ju varit lite samtidigt.... man kan säga den här ISO9000 vi fick var ju också i samband med... då omorganiserade vi också en hel del, i samband med det som man körde det också då. Det gjorde att det blev en hel del förändringar då. I viss mån när vi omorganiserade och gick över till Java, blev det också en hel del förändringar. Men det är också...det var ju ett teknikskifte som gjorde att man ändå var tvungen att göra förändringar. Någon stor förändring just i metoder som vi gjorde...jag vet inte om det hade med omorganisation att göra...men det var ju när vi började med standardiserade utseenden på...på....redan på våran RPG-version då. Vi sa att ett program av den här typen skall se ut så här och jobba så här och den här typen skulle det vara, innehålla dom här delarna.

B: Jag har en fråga, om jag får lov....jo...ni nämnde förut att under den glada tiden, före 2000, när det dök upp en väldig massa nya människor, var det ju även aktuellt med en del företagsköp. När Intenia köper upp ett företag, till exempel utomlands, vi säger Asien. Nu vet jag inte om det har förekommit, men vi tar det som exempel. Köper upp ett företag på 20 man eller så som är vana vid..."jaa...vi kör enligt den här systemutvecklingsmetoden". Dom blir ju tvugna att infoga sig i ledet, så att ni kan prata med dom. Hur går en sån process tillväga, för det är ju inte speciellt enkelt...finns det en pärm som man slänger i knät på dom och säger att: "nu gör ni såhär istället"? Det är ju en sak när man har ett gäng nyutexade personer eller som kommer nyanställda så kan man köra dom som traineer ett tag, med mentorer och hela kittet, men det är ju lite svårt när dom är ganska många mil bort på en annan kontinent.

P: Men...i princip slänger vi ju en bunt med grejor i knät på dom och säger: "såhär skall ni jobba".....för vi har ju en modell...ja..det som vi köper in och som liksom...det är ju "business partners"...dom går ju inte direkt in i vår utvecklingsverksamhet utan det är ju såna som är konsulter som skall sälja och implementera då....

B: Jaha...ok.

P: Dom får ju...där har vi ju metoden Implex[implementationsinstruktioner för applikationskonsulterna] som är...den finns där den är, den skall dom följa. Sen kan det ju ta en liten stund innan dom kan den, kan följa den. Sen ställer vi ju oftast, jag vet inte, ställer vi ju upp med folk från Sverige som hjälper till från början, som jobbar där och då styr man och hjälper till att komma in i det.

B: Men utvecklingen sker i första hand i Sverige, eller?

P: Ja.

P: Men vi har ju köpt upp bolag, och inkorporerat där också, men då blir det ju lite annorlunda. Då har det ju blivit en mix, vi har ju tagit in...när man får in det folket så tar man ju in nya grejor där också då. Så då har det ju liksom mergat ihop sig.

M: Kan man säga så här egentligen då, att det är deras...dom har ett gäng verktyg som dom är vana vid, inte bara typiska verktyg, programvaror och sånt, utan lösningsverktyg som dom har i huvudena och när dom finns med i era projekt och så, så delar dom med sig, så eran erfarenhetsbank och verktygslåda blir större.

P: Där försöker man ta vara på, komma fram till vilket sätt är bäst, hur skall vi jobba? Då har det liksom blivit en hopmergning då man försökt ta vara på det som är bäst. Som vi upplever som bäst, som funkar bäst för oss.

T: Det är nog lite...det är jävligt hårt...det är Implex, våran projektmodell som gäller nu som...ute på konsulterna är det definitivt det. Det är där man kanske har tagit mera.....som jag har hört liksom mest, störst skillnad det blir ju sättet man säljer in. När vi skall försöka få det sålt, där går det ju inte bara att.... sättet man säljer på i Japan kontra Sverige exempelvis, är ju väsensskilt, där bygger det ju på traditioner och nu är det säkert föremål för förändring också, men har du väl fått en kund där så har du honom, men att få en kund kan ju ta årtal och liksom det fungerar ju inte här då. Där är det ju större, tillåts det av naturliga skäl större skillnader och sen när väl projektet, försäljningen är i hamn så skall det ju drivas på ett likartat sätt. Sen vet jag inte hur många utvecklare vi har men dom har varit mest norden-baserade, nästan uteslutande.

B: Men Implex är en projektstyrnings....

P: Ja, det är implementering av Movex-systemet ute hos en kund då, som konsulterna jobbar... som innehåller då alltihopa med att sätta upp det, dokumentera, utbilda, anpassa och ja...allt som behövs... i alla dom stegen... processen som behövs... från att man startar till drifttagande och support efteråt.

T: Som egentligen bygger på att man skall kunna upptäcka avvikelser tidigt i kedjan så att du inte skall behöva iterera tillbaka...max ett projektsteg så att du inte skall behöva "gå tillbaka till gå" och att du tillsammans med kunden alltid skall gemensamt godkänna ett delområde i projektmodellen innan du går vidare till nästa.

B: Upplever ni att ni på IRD behöver eller skulle kunna ha nytta av att ha en såpass styrd metodik som Implex? Eller skulle det bara vara i vägen för er, för ni bryter ju ändå lite ny mark...?

P: Njaj, fast vi är och andra sidan ganska hårt styrda av ISO9000, den är ju lite motsvarande för vår del, den processen och vår standard styr ju mycket.

B: Och med standard menar du kodstandard?

P: Ja, både det och standard för olika typer av program och hur det skall se ut, kommunikationen i mellan och... interface och allt sådant... alltihop är ganska hårt standardiserat då, "så här skall det vara"... Det får man inte avvika ifrån, så här skall det vara, det styr ganska hårt...

B: Har vi fått med dom frågorna som vi hade?

M: Ja, det tror jag nog att vi har...

Bilaga 6, Intervjumaterial Accenture

Transkribering, Accenture, 2004-04-19

[...]=efterhandskommentarer av intervjuarna

M=Mattias Grytting

B=Björn Olsson

P=Accenture

B: Hur skulle du beskriva Accenture?

P: [Plockar fram presentation] Det som vi vill utmärka oss med....vi är inte bara en...vi vill inte leverera ett system som kunden beställer utan vi vill leverera nåt slags värde till kunderna, där man liksom först tittar på processer och affärsnytta och har ett tydligt business-case så man kan motivera den här systemimplementeringen. Det är ganska sällan eller ogärna som vi bara levererar ett system som dom har specificerat. Förr eller senare då så stöter man på patrull någonstans i organisationen, nån ifrågasätter vad det är. Ofta är det...ett typiskt projekt skulle vara att vi gör...kanske att vi gör en strategifas först, så man kollar vad man skall göra, sedan en organisatorisk...en processdesign....”jobba såhär istället”...och så försöker vi bygga en systemleverans som stödjer det. Och inte tvärt om. Ehh...ja....bla...bla...[studerar sin Powerpoint-presentation]. Det sättet som vi är organiserade på....dom lodräta staplarna är “global market units”...”operating groups” hette de förut, ni ser att det står så här[pekar]. Ehh...som då är organiserade efter klienternas industrier. Vi jobbar med “communication & hi-tech” där Ericsson, Telia är typiska stora kunder. “Financial services” är banker och försäkringsbolag. “Government” har varit ganska litet i Sverige, men det är då regeringar och institutioner. “Products” är den som är störst i Sverige just nu, “communication & hi-tech” var störst innan it-bubblan sprack. Nu är “products” störst och speciellt i Göteborg, det är alltså tillverkande företag, som Volvo och SKF, Mölnlycke....och “resources”...olja, gruvindustri, även energibolag. Den har varit väldigt liten i Sverige, stor i Norge, men sen elmarknaden släppts fri har den vuxit i Sverige också..för alla elbolag räknas in här. Sen i andra dimensionen har vi då [pekar på Powerpoint-presentation] våra kompetenser...eehhh....strategier, “crm”, “scm”, “performance”, “finance and performance management”. Jag hör till den här då, “solution engineering”, som är den tekniska delen. Hela den här är liksom om konsultverksamheten [pekar], den ursprungliga delen av företaget. Här kommer då den här [pekar], som är nytt egentligen. En strategisk leveransmodell där vi har det här...vi vill liksom ha branschkunskapen och vara lokalt närvarande hos kunderna. Vi har, på två nivåer egentligen, de globala utvecklingscentrerna och även på en närmare nivå....att vi kan göra det här i Spanien till exempel. För att det här skall vara lyckat så måste det vara ett ganska stort projekt. Det är inte alla projekt som är så stora att det lönar sig att lägga utvecklingen på andra sidan jorden, det finns uppenbara nackdelar med det också. Dom nackdelarna blir större ju mindre projektet är. Ehh...ja....kanske inte[bläddrar vidare i Powerpoint-presentationen]....den här kan vara intressant [pekar] på ett sätt också...det här kanske är en av anledningarna till att utvecklingen...systemutvecklingsmetoderna ändrar sig i det långa perspektivet, att...om vi på 80-talet var en systemleverantör, som levererade ett system till ICA, som var en stor kund på 80-talet, eehh...så har det gått...ja...kunderna frågar efter större och större åtaganden i tid. Man vill inte bara ha en komponent eller ett system eller någonting utan man vill att vi utvecklar, levererar och driftsätter någonting, men även sköter driften och det sista steget är ju nu “outsourcing” där dom till och med vill att vi tar över delar av deras personal för att ja... driva det här i ett antal år framåt. Det är en väldigt tydlig trend att utvecklingsuppdragen blir större, färre och löper över längre tid framförallt.

Och det i sin tur är kanske eller....ÄR en av orsakerna till att...Accenture var tidigare partnerägt det var inget publikt företag som handlades på börsen utan det ägdes av partners och lite förenklat så kan man väl säga att vid varje årsbokslut så tog man pengarna man tjänat och delade ut dom... Sen två år tillbaks så handlas Accenture på New York-börsen och anledningen till det var att vi hade inte råd att konkurrera med sådana här jättar som IBM, som är största konkurrenten. Eftersom projekten blir större och längre så blir det också mycket dyrare införsäljningsprocesser. Det kan kanske ta två år innan vi börjar tjäna pengar på ett projekt och sedan tjänar man pengar på det i åtta år. Då hade vi inte råd att IBM vann för att vi inte kunde göra ett projektutlägg där man behövde betala....

M: Dom hade den finansiella uthålligheten egentligen...

P: Precis. Så nu är vi publika och har fått in kosing till det... Att vi inte bara levererar system utan vi vill koppla det till ett värde [tänker högt och bläddrar i Powerpoint-presentationen]... Konsultverksamheten är historiskt sett den verksamheten som vi kommer ifrån, sedan har vi då byggt upp det här i en egen teknikorganisation runt omkring och även allianser med strategiska partners, framförallt tre stycken, det är Microsoft, SAP och Oracle. Det är dom vi är kompisar med...och HP då på hårdvarusidan. Våra konkurrenter då är IBM som har både hårdvara och konsulting själva. Ehhh.... ja vi är 83.000 personer. "Outsourcing" är stort och nytt och det tror vi mycket på. Allianser för att projekten blir så stora och kompletta, man frågar efter kompletta lösningar och vi kan inte äga alla utan det blir dom här allianserna, att det bildas grupperingar och det här då...jag som inte har varit här sedan jag slutade skolan utan kom hit efter arbetslivserfarenhet slås ju väldigt mycket av det här att jag kan ringa någon på andra sidan jorden och man pratar ändå samma språk på ett sätt så att man kan sätta ihop ett projekt. Jag vet att till den här rollen behöver jag en "consultant C3", vi har en sådan här skala på människor när dom avancerar i karriären. Och det betyder samma sak på Filippinerna som det gör här och det är imponerande att det är samma tidsredovisningsystem där som här och det är väldigt standardiserat på både ont och gott. Det blir inte så lokalanpassat men...styrkan är just det.

B: Det är ganska intressant, för det blir ju lite som...Accenture är ju så fruktansvärt stort, men om man jämför mindre företag, så när dom skall samarbeta i systemutvecklingsprojekt så blir det ju en klar fördel att RUP är så stort, för den har ju klara rollbeskrivningar och alltihopa. Det är ju precis samma sak som du säger, fast här pratar vi om ett företag och dess kultur istället för en systemutvecklingsmetodik.

P: Ja...och...vi har ju den här...vår systemutvecklingsmetodik också, som inte alls är olik RUP på något sätt...eehh....den har hetat olika saker under åren här..."Method One", under 80-talet, "BIM", "Business Integration Model" under 90-talet, och nu, det är mindre än ett år sedan, har man döpt om den och kallar den för "Accenture Delivery Suite". Den största skillnaden...det är den sista...det kan ju vara intressant för er då...när man gick från "Method One" till "BIM" var det framförallt att man ville ha ett större verktyg som inte bara täckte systemutvecklingen, utan man ville även ha med den här strategi....utveckling, driftfas så att man fick ett mycket större perspektiv. Den stora ändringen nu, från "BIM" till "Accenture Delivery Suite" är att den består av olika moduler och där vill man då ha...vissa moduler är förberedda för dom här standardverktygen, så det finns till exempel en SAP-modul - "så här implementerar vi ett SAP-projekt, så här implementerar vi olika Microsoft-grejer", det går mot mer...man skall på nåt sätt tvingas välja några partners för att kunna leverera nånting snabbt, till priset av att vi inte längre kan kalla oss oberoende konsulter. Man skall nog inte säga att man skall ta in oss för att avgöra om man skall använda IBM- eller HP-grejer, för då

hade vi svarat HP. Ehh...vilket vi kanske inte hade gjort för 10 år sedan. Man har bedömt att det är viktigare att vi har färdiga lösningar och snabbt kan leverera nånting som är nästan exakt det man vill ha, snarare än att man levererar exakt det kunderna vill ha till ett mycket högre pris och senare i tiden. Där har man valt väg då, vi jobbar helst med Microsofts produkter och HP och SAP.

M: Vilket för oss osökt in på våran första egentliga fråga, om man skall kalla det för det. En metod, en systemutvecklingsmetod kan ju vara väldigt mycket olika för olika personer och olika företag och så. Man kan se på det på väldigt många olika sätt, på många olika nivåer. Om du skulle beskriva litegrann: vad är en metod för dig och ert företag, och i samma andetag fundera litegrann på; vad är en god metod kontra en metod som inte är god, vad är det som skiljer en metod från att vara bra eller dålig?

P: Mmmmm....vad det är...jag ser att...det är ju ett sätt att överhuvudtaget göra det möjligt att sätta ihop stora internationella projekt, just att ha ett gemensamt språk... att när folk säger intergrationstest så menar man samma sak både här och på andra sidan jorden. Det är väl en uppenbar anledning till att använda en... Men även liksom en slags.... det är ju en kvalitetssäkringsmetod... det ger individerna eller de individuella projekten en...vad skall man säga...en ledstång att hålla sig i. Det finns nog ingen som är expert på alla områden, utan man går till metoden och ser "vad är det jag skall göra i den här fasen". Vad är det mer... det är väl kanske också ett... typiskt i ett projekt att man kanske... vi bestämmer att.... vi vet inte riktigt vilken metod vi skall använda men vi börjar med den här, den är ju säkert tillräckligt bra. Då har man en utgångspunkt, man börjar med den metoden och sen ser man att, "njae...95% passar bra, men sen så gör vi så här istället". Den här delen av metoden kan inte vara gjord för det här projektet eller vi skall inte ha den komponenten. Det är liksom ett sätt att slippa börja jobba med blanka papper och uppfinna hjulet igen. Vilket är en konkurrensfördel då, för hade vi gjort det så hade ingen köpt projektet för det hade tagit för lång tid. Då blir det för dyrt. Bra och dåligt? Jaaaa... om man ser till varför dom har ändrat sig då så är det väl just för att ... dom tidigare inte har hängt med i affärsklimatets utveckling. Man har gjort en metod som passade på 80-talet men nu är det ingen som köper den typen av projekt längre. Man vill ha mycket större...dels köper man större projekt, längre tid, och då blir det en annan metod eller metoden skall passa för det. Och sen det här att den är tillräckligt stor att den täcker in alla... inte bara systemimplementationsdelen bara "design, build, run" för ett system utan man har fler saker runt omkring, projektledning och programledning om man har ett stort program med flera projekt... Att man tänker på hur det skall driftas så småningom, att man får med det tidigt. Att den är modulär på det sättet... det är bara dom allra största projekten som behöver använda allt.. kasta boll kanske[otydligt]... Strategidelen av ett projekt eller den sista[del]en..."okej nu har vi gjort en egen design här nu får ni ta vid". Att man kan använda valda delar av den.

M: När vi satt och funderade lite grann så försökte vi rita någon form av konceptuell bild av hur vi såg på det här med metoder och så här... Och då kom vi fram till den här bilden egentligen, som representerar vår lilla egen modell på hur vi tror att det här fungerar då[plockar fram vår förklaringsmodell]. Då sa vi så här att vi kunde se tre stycken nivåer, vi kunde se en strategisknivå, en taktisk nivå och en operativ nivå och dom här nivåerna korresponderar egentligen mot det som vi valde att kalla för process här. Den strategiska nivån, här handlar det om...vad skall man säga... en metod eller metodologi eller process som gör mer än bara systemutveckling. Lite grann det som du var inne på. Alltså, här finns också sådana saker som styr projektledningen....

B: Där sägs egentligen vad det är som skall göras men inte hur.

M: Precis. I vilken ordning...vilka grova faser som finns, hur ett projekt skall dras upp, riktlinjer för...och så vidare. Inom den här processen så på den taktiska nivå så finns det en eller flera metoder eller metodologier som säger att: "ok..i fas nummer vad det nu kan vara, så skall vi faktiskt"...den skall inrymma de här sakerna. "För att göra de här sakerna så använder vi den här metoden, för att fixa det." Den som finns på den taktiska nivån.

P: Du menar att man skulle ha olika metoder.....?

M: Ja..det är fullt möjligt alltså. Se det som ett litet ramverk om man säger så. "Det här är vår...den här talar om vad vi skall göra, men inte hur [strategisk nivå]"...och "den här talar om hur vi skall göra dom enskilda sakerna den här [strategisk nivå] har talat om att vi skall göra [taktisk nivå]". "Så här betar vi oss för att göra detta". Sen på den operativa nivån, och nu talar vi skruv och mutter alltså, så vet kanske...om vi tar ett väldigt enkelt exempel, en metodologi säger så här: "I den här fasen så skall vi till exempel...den här fasen består av följande aktiviteter: vi skall rita ett klassdiagram" Detta säger metodologin. På den operativa nivån säger aktiviteten här [pekar], den säger hur vi skall rita, hur vi skall komma fram till det klassdiagrammet, till exempel. "Vi gör det med hjälp av UML" till exempel, eller med hjälp av utvalda delar av kundens organisation som får vara med och jobba fram klassdiagrammet. Så här såg vi det, att det skulle kunna vara uppbyggt. Om jag har förstått dig rätt så när du talar om Accentures metod, så talar vi egentligen om den här [pekar på strategisk nivå] delen. I Accentures metod...den styr egentligen alltifrån projekt till verksamhetsutveckling, men den rymmer också saker som talar om hur man skall bedriva systemutveckling, kanske inte på skruv- och mutternivå men...i vilken ordning saker och ting skall komma.

P: Joo.....den är väldigt omfattande på så sätt att den har allt här [pekar] den har både den här...som ni kallar strategisk nivå, men du kan även borra dig ner till skruv- och mutternivå. [plockar fram en hemsida med metodinformation från Accenture]. Nu skall vi se....."Accenture Method Library"...nu blir det lite improvisation här....men med lite tur så....

Här har vi kanske det du kallar strategisk nivå...nu är det här ganska nytt då...företag använde vi "BIM" då, men här har dom en lite annan nomenklatur...men här har vi typiskt...."BI" här då, "business integration"...med hänvisningar tillbaks till....den gamla metodologin, som dom ännu inte har översatt här...sen har vi utvecklingsfaser där du har...och det här är nytt då....här finns det....

M: Ja just det...här har du dom grenarna[instanser] som du nämnde tidigare.....

P: Här finns ett specifikt nu, för SAP, en annan "package"...vi kan ta en sån här då som....

B: Du hade någon metod där för integrering med Peoplesoft.

P: Vi kan titta på den då...

Det gick inte nej, den behövde Lotus Notes.

Den kommenterar lite grann då kring "project management", om man tittar på technical architecture i dom olika faserna då... om man vill borra ner sig i analysfasen[klickar ner sig].

B: Som jag ser det här så är det nästan så att er metodologi ligger runt hela vår modell och har flera sådana här[pekar på vår modell]. Det här är bara systemutvecklingsmodellen sedan har ni en likadan för...

M: Verksamhetsmodellering eller processutveckling.

P: Ja just det, en annan dimension här, så kan man ju se det... ja just det, för det är ju inte så att man skall använda båda utan det är ju en färdiggjord... Egentligen är väl det bara specialfall och genvägar till den här generella utvecklingen. Jag tänkte jag skulle borra ner mig här...

B: Det är ju jätligt intressant, men det är ju för att Accenture är så stora också, att ni har så vitt skilda... att ni har så bred verksamhet...

P: Se hur djupt ner man kan komma här... "solution blueprint" [tänker högt och klickar neråt]...

B: Här har ni ju till och med templat och exempel.

P: Då fyller man i den snabbt så tror kunden att man har gjort det från början och betalar för det.

B: Det var då jag skulle stoppat bandet, du får ge mig lite förvarning så jag hinner göra det.[skratt, alla]

P: Ja just det, men det är väl mycket av hemligheten, det är ju ingen hemlighet egentligen men att man inte uppfinner hjulet gång på gång. Att ha det här stödet att ha dom här metoderna. Så på det sättet kan man säga att metoderna är levebrödet på något sätt. Varje intelligent människa hade säkert kunnat komma fram till det här själv men då hade det tagit oändligt mycket längre tid.

B: Det hade inte sett ut på samma sätt heller.

P: Nej, just det.

M: Om man skall ta ett praktiskt exempel här...om vi tänker oss en situation att.. om vi tar exemplet med Kungsbacka om du är med där och koordinerar i ett större projekt och därifrån tar du med dig sådana saker som du till exempel skickar till Spanien och så säger du att: "det här vill vi ha gjort" rent systemutvecklingsmässigt, "skall ni ta och hantera det". Du jobbar ju i det projektet efter en sådan här styrmetod eller vad man skall kalla det för så att du vet vad som skall göras och så, eller?

P: Ja det här är ett problem då när vi inte har hela projektet själva, för nu har vi då bara en del i ett projekt där någon annan har gjort den stora bilden och använder sin terminologi och sin metod som inte vi kan. Där har vi redan...vi har bara varit där en månad men redan har vi sett begreppsförvirringar, dom säger en sak och vi tror att dom menar något annat för det är inte så vi kallar det. Det är ju kanske tvärtom då, det är väl ett bevis på att en universell metod eller en metod som alla kan i alla fall gör nytta. Här ser man vad som händer när man inte har den. Det finns ju andra bra anledningar till att göra som vi gör, men detta är ju en nackdel, att inte ha...alla i projektet talar inte samma språk.

M: I vilket fall....och det har vi konstaterat...så kan det vara. Om vi nu tittar på den delen som ni gör, som Accenture gör så...tar du med dig saker därifrån som skall göras och så fixar du i ordning det så att dom i Spanien förstår vad dom skall göra. Sen skall dom göra detta. I vilken utsträckning styr...jobbar ni internt efter eran metod...och isåfall i vilken utsträckning styr den metoden spanjorernas arbete? Alltså...spanjorerna får förmodligen någon typ av kravlista från dig eller nån typ av sådan....vad det nu kan vara...skiter du i hur dom gör det, bara dom gör det....eller....är det viktigt....gör dom det alltid på samma sätt, för att man skall kunna spåra....?

P: Det dom gör....det dom gör internt, det gör dom väldigt strukturerat. Dom är väldigt.....ehh....både för och nackdelar med det...men dom är väldigt strukturerade, jobbar enligt metoderna. Inte nödvändigtvis bara våra metoder, utan här är det SAPs, "standard procedures". Det finns något som heter "LFMW", som är ett verktyg för att göra "upload tools"...migreringsverktyg. Det är ingen Accenture-standard, utan det är en SAP-standard. Denna följer dom väldigt slaviskt och...fördelen med det är ju att...dom har ju blivit experter på det här, dom kan ju utnyttja gamla erfarenheter i Spanien. Nackdelen är att dom är lite för...."oj...vi har fått ett krav som inte passar in här...."....så blir dom liksom handlingsförlamade..."vad gör jag nu då? Sådär skall det inte se ut. Enligt metoden skall man göra såhär och det här och det här". Kolumnerna är inte i rätt ordning....det är en nackdel....dom är väldigt strukturerade, och det måste dom nog vara för att kunna jobba på distans...dom har ju ingen känsla för vad kunden har....jag är tvärt om då, jag lever med det här dagligen, jag förstår vad dom vill.

M: Medans dom får sitt på posten mer eller mindre...

P: Precis. Men jag kan inte, å andra sidan, metodologin utan och innan och i ryggmärgen, som dom kan. Det är väl det som blir min uppgift, liksom, att försöka översätta det här....eehhh...om dom liksom bara nämner att: "vi skulle vilja att ni tittar på den här abap-koden också"...försöka få in det då i den här väldigt strukturerade organisationen.

M: Om man skall försöka sig på nån sorts slutsats här då, kan man säga att avstånd i det här fallet, geografiskt eller....att det skulle kunna vara en hämmande faktor i till exempel anpassningssyfte...att ju längre avstånd du har, desto svårare har du att anpassa din metod?

P: Anpassa metoden....för....

M: Ja, alltså, dom håller sin metod väldigt....står en kolumn fel så får dom fundera på hur dom skall hantera det här. Det beror kanske delvis på att det finns ett långt geografiskt avstånd där man inte har kontakt med kunden, där man inte kan justera själv, så att säga, utan man får ett mer eller mindre färdigt material som man skall jobba med.

P: Det ligger väl litegrann i hela...eller 'ja', då för att svara kort. Dom får ha sitt standardiserade sätt att jobba på och dom kan lägga mycket folk på det där. Ehh....det blir ju inte lika effektivt som om samma antal människor hade suttit här, på plats och gjort det.

M: Men det var ju inte det man ville heller....

P: Precis, men totalen blir bättre även om man har mindre effektivitet räknat per person så har dom väldigt mycket erfarenhet. Dom har gjort ungefär samma sak 100 gånger tidigare och då har en betydligt lägre timkostnad.

M: Det som hamnar inom metodens ramar går så mycket fortare så att man kan ta dom grejerna vid sidan om och lösa dom, fixlösa dom...

P: Och fördelarna är större än nackdelarna, men visst finns det nackdelar... men där skulle jag vilja säga att min erfarenhet är att språk och kultur är större barriärer att komma över än det geografiska avståndet för att på ett sätt är det lättare att jobba med japaner än spanjorer kanske.

B: Italienare och deadlines har jag fått höra en del om i alla fall.

P: Det är ju mer en rekommendation för dom...

B: Jag har en fråga då som jag kan dra med en gång. Ni har ju er egen metod för hur systemutveckling skall gå tillväga, sen nämnde du att ni körde något projekt ute på Volvo och då nämnde du att ni körde RUP. Är det så att konsulterna inom Accenture kan ett flertal olika metodologier eller var det så att ni hux flux fick lära upp folk i RUP?

P: I det fallet var det så att vi jobbade tillsammans med Volvo IT. Det här var lite grann innan vi hade dom här "delivery centers", innan det blev stort så att säga. För att klara av den leveransen så gjorde vi det tillsammans med Volvo IT och dom var så att säga våra systemutvecklare och då bestämde vi att vi skulle använda RUP, RUP var den standarden som Volvo IT använde på den tiden och kanske gör fortfarande. Jag tror att det var ganska nytt då, för då gick vi från Accenture en "crash course" i RUP men det vet jag att det var många från Volvo IT som var med på dom här utbildningarna i början, så att det var en utbildning i det här.

B: Men hur skulle det fungera idag om ni skulle göra någonting och dom absolut vill att ni skall köra RUP eller att ni skall samutveckla något till exempel med Volvo IT igen, nu kanske inte det skulle vara aktuellt, men vi leker med tanken. Finns det kunskap om RUP inom organisationen nu eller...?

P: Nu finns det ju det på grund av eller... vi har ju erfarenhet av kundernas system eller metoder. Vi jobbar till exempel mycket med GM, inte i Sverige men runt om i världen. GM är så stora att dom har sin projektledningsmetod. Och där har ju vi lärt oss det genom att jobba med dom och nu ställer dom krav på att vi och andra som vill bjuda på jobb kan deras system och det är alltså ingen global...alltså, det är ingen extern standard som RUP, den är det ju Rational som har tagit fram, men dom är ändå så stora att dom kan kräva att konsulterna kan deras metod. Så där vi kan det så beror det ju på erfarenheten. När man alltså börjar på Accenture så finns det inte ett paket med kundernas interna systemutvecklingsmetoder som man får lära sig utan man lär sig vår metod, och när det blir dags för ett projekt så ser vi att: "ja okej...här behöver vi fem man, två av dom som finns tillgängliga har redan jobbat på ett projekt med det här så det blir ju jättebra dom tar vi direkt dom andra tre får vi skicka på snabbutbildning" eller så att säga acceptera någon slags... dom kanske är med i projektet men vi tar bara betalt för halvtid dom första två månaderna för att vi vet att dom kommer inte vara effektiva eftersom dom inte kan det här men dom lär sig det med tiden. Så att...ja...absolut att vi måste och vill använda andra metoder än vår egen, vi kan inte säga att vi vägrar om vi inte får använda vår egen. Men....utbildningarna och det görs liksom när det behövs.

B: Så ni ser det inte som ett problem att ni inte har RUP som ett universalverktyg eller så?

P: Neej...det tror jag inte. Min erfarenhet är att det kanske inte är så viktigt om man väljer RUP eller om man väljer våran metod. Har man lärt sig en metod så finns det större likheter än skillnader och det är just att använda metoder som är det viktiga. Sen kan ju RUP i många fall vara enklare om den passar in just till det här. Det är möjligt att vi skulle välja att föredra att använda RUP om det är ett projekt som passar just för RUP och vi har använt det tidigare.

M: Får det faktum att ni, villigt eller ovilligt, fått en del kunskap inom organisationen om till exempel GM:s övergripande metoder för projektstyrning och så...får det konsekvenser för era metoder? Inkorporerar man bra saker ifrån GM:s metodik i eran egen och säger: "dom gör det här riktigt smart, alltså...vi tar den och lämpar in den för det är tjosan"?

P: Det gör vi. Vi har liksom en...att underhålla en sån här intern...standard kräver ju en organisation i sig...så här finns det...nu vet jag inte exakt hur den ser ut, men...det finns uppenbarligen många personer som har till uppgift att underhålla...inte bara våra metoder utan våra interna kunskapsdatabaser, "det här är det vi har lärt oss från projekt tidigare". Det vill ju vi göra tillgängligt för kommande projekt. Det finns en administration runt det där som måste finnas för att det skall fungera. Dom måste ju på nåt sätt...det finns ju metodexperter då, och dom plockar ju säkert med sig erfarenheter från kundprojekt. Det är kanske det bästa sättet att lära sig en metod...att jobba i projekt med den. Utan att veta exakt hur det går till...alltså hur dom här som ansvarar för att en sida som den här finns [pekar på metodhemsidan] så...ja...det gör vi.

M: Det var ganska intressant, för det var en av dom anledningarna som...när vi pratade med ADB-kontoret, så hade dom ju...dom har ju varit kommunalt bolag innan och jobbar mycket med kommunens system och så och dom har haft en egen systemutvecklingsmetod som dom har använt i många år. Men nu har dom bytt, så nu kör dom RUP, och ovanpå den så har dom lagt en egen projektstyrningsmodell. En av anledningarna till att dom valde RUP var att dom tyckte att dom inte hade den organisationen som krävs för att hålla en egen metod uppdaterad och levande, igång hela tiden. Det behövs så stora organisationer och det är sådan overhead på det så att dom klarade inte det. Dom har valt att låta Rational utveckla RUP och så tar dom den vid sidan om.

P: Det låter ju vettigt....om jag gissar...man måste tillhöra dom största där för att...

M: ...ha något utbyte av att bränna den energin på att hålla sin egen metod vid liv...

B: ADB-kontoret är ju tämligen små...300 anställda....

P: Sen vet jag inte var gränsen går där...

M: Nä...men just diskussionen var ju intressant....

M: Då skall vi se....ja...lite abstrakt fråga kanske.....vad är det som driver evolveringen av er metod, vad är det som gör att ni ändrar i den? Vi har varit inne lite på det förut med tidsperspektivet och längre uppdrag och så. Går det på något sätt att mappa till vår bild här? Vi sa att på den här nivån så finns det ett antal faktorer som kan medverka till att man anpassar det här[strategisk nivå] och här är det sådana saker som omvärldsfaktorer. Vad gör våra konkurrenter? Till och med sådana faktorer som trender inom branschen som säger att:

“nu är det hett att göra så här”. RUP kan vara ett exempel på det, det är och har varit trendigt att nu under några år använda RUP.

B: DSDM är väl på väg nu också.

M: Ja. En dynamisk utvecklingsmetod. Den typen av faktorer befinner sig på den strategiska nivån. På den taktiska nivån så befinner sig faktorer som, hur vi är organiserade? En metodologi, en metod, en metodik kan ju antingen anpassas till den befintliga organisationen eller så tenderar det att vara så... och nu spekulerar jag lite vilt här... att när man har bestämt sig för att vi skall göra en omorganisation så tittar man också väldigt ofta över metodologin och så säger man så här att: “ok vi anpassar organisationen efter vår ny metodologi istället för att anpassa metodologin efter organisationen”.

P: Jag skulle tro att den största drivkraften här uppe, den så att säga långsiktiga utvecklingen[strategisk nivå], det är så att säga branschens utveckling. Det här med att projekten blir större och större, det kommer “outsourcing” in i bilden, vi tvingas till offshoreverksamhet och då hänger liksom vår metod som vi hade...hänger liksom inte med då. Det fanns ingen beskrivning om hur man hanterade 100 programmerare i Indien. Så att omvärlden förändrar nog, eller tvingar fram en förändring av vår organisation eller av vår affärsmodell. Vi kan inte vara konkurrenskraftiga längre om vi bara bemannar projekten med svenskar eller bara tyskar. Så omvärlden driver nog fram en förändring av organisationen, som driver fram en förändring av metoden. Och sen så är väl även en annan faktor att på den operativa nivån, i och med att teknikutvecklingen går framåt så måste vi ha fram mer detaljerade metoder för nya programmeringsspråk, nya utvecklingssätt, nya... ja nyheter som kommer från tekniken och inte från... ja dom hänger väl ihop på något sätt men ja... det är svårt att säga vad som är hönan och vad som är ägget. Men teknikutvecklingen tror jag kan driva metodutvecklingen underifrån.

M: Det är nog inte helt otroligt nej.

P: Det är ju bara mina egna observationer här, men...

B: När vi ändå är inne på anpassning av metoder, hur går er anpassning tillväga? Du nämnde att ni i början på ett projekt såg över vad projektet behöver. Återigen om vi kör RUP då så har dom ju någonting som dom kallar för “development case”, då man så att säga tittar på... man går in i början av projektet och tittar på vad är det här för typ av projekt, vilka “deliverables” kommer vi behöva? Vilka delar kan vi plocka bort. Är det lite så som ni jobbar också eller...?

P: Ja, både och. Den här långsiktiga utvecklingen av metoder, den pågår över huvudet på normala människor... långsiktig... det kanske dyker upp något helt nytt var tionde år som det precis har gjort nu då. Medans den här lilla lokala anpassningen är från projekt till projekt. Om... nu vet jag inte exakt hur mycket RUP täcker, men säg att vi har en.. säg att RUP även täcker driftfasen men att vi har ett projekt där vår leverans är en kod och kanske support vid driftsättning men sen kanske inte vi är med längre. Eller om vi kommer in där strategifasen redan är gjord, det är inte längre nå...business caset finns redan, liksom...”vi skall göra det här för att minska de administrativa kostnaderna”. Då kan vi plocka bort dom delarna från metoden som stöttar det. Dels det...och sen...min erfarenhet är att vi under projektet typiskt inser att: “oj...här går tiden...vi måste göra så här...visst vi vet hur viktigt det är att skriva detaljerade testcase...samtidigt som vi skriver use-casen...fast det hinner vi inte...vi gör det lite slarvigt nu, så tar vi den smällen sen”. Att även projektledningen tar dom besluten. Det är väl

också en anpassning av metoden, metoden säger ju att vi skall skriva testcase samtidigt som...han testledaren är mammaledig [sic!] eller nåt sånt där. Mer dag till dag-anpassning. Eller kanske inte en anpassning, kanske mer ett avsteg ifrån.

M: Fast det är ju å andra sidan också en anpassning, utan tvekan...en sorts anpassning. Som vi ser det så...en god metod ger väl möjligheter att göra den typen av anpassning också, alltså att man plocka bort saker som man inte: a) hinner, b) tycker är tillräckligt viktiga. Det är en nog så viktigt del av en metod att den är uppbyggd på ett sådant sätt att man klarar av att göra den typen av anpassningar.

B: En metod täcker ju normalfallet.

P: Annars kan jag tänka mig också att metod är litegrann till för att få projektet att inse vikten av att INTE göra för stora avsteg, att man...utan metoden hade man kanske inte tänkt på att jag kan vinna mycket på att skriva testcasen samtidigt som jag gör use-casen, även om jag inte skall göra testerna förrän om 3 månader. Så det är väl en av fördelarna med metoderna, att göra det där klart för projektet.

M: Tänkte på en annan sak...låt oss säga nu...Accenture är ett väldigt stort bolag...det innebär att man förmodligen, antingen har eller gör.....växer med köp av andra företag eller konsultorganisationer och så vidare...integrerar dom i sin verksamhet. Hur går man till väga när man...låt oss säga att man köpt ett företag nånstans...som har en systemutvecklingsverksamhet. Hur talar man om för dom här människorna att: "nu ingår ni i Accenture, vi har ju en metod som vi använder vid uppdrag." Det uppköpta företaget har ju förmodligen också redan en metod. Hur gör man övergången där?

P: Jag kan säga som såhär: Accenture har inte vuxit på det sättet, utan har vuxit organiskt och växt...det finns inga fall man kan titta tillbaka på, "när vi köpte den"...som Cap Gemini när dom köpte Ernest & Young. Däremot en av våra...saker som vi jobbar med inom konsultingdelen, just det här "post-merger-integration". Vi hjälper företag som är just i den situationen. "Ni har köpt det här företaget, ni skall gå samman, hur gör man det här smidigast?" Men det har jag inte jobbat själv med, dels är det ju en hel vetenskap i sig...men det är väl...man skall ju åtminstone ta hand om process-integrationen, alltså att få dom att arbeta på samma sätt i varandras processer. Sen kan det ju variera...om man köper nånting för att förstärka eller förstora en befintlig organisation, eller om man köper ett företag för att bredda sig själv, man köper kanske sin distributör eller sin leverantör....och sen den tekniska, data-integrationen. Så det är absolut nånting som Accenture som företag har väldigt mycket erfarenhet av att göra åt andra, men ingen erfarenhet av att göra själva.
Nu glömde jag av frågan.....

M: Ja, anta nu att ni köper ett företag, hur talar ni om för dom att nu skall ni använda den här metoden. Men eftersom det inte har hänt så är det ju lite svårt.

P: Den får jag nog passa på, dels för att vi inte har gjort det själva och dels för att jag inte har jobbat med den delen av Accenture som gör det.

B: Är det tänkt att era metoder skall anpassas eller skall dom användas strikt? Vilken frihetsgrad finns det så att säga?

P: Vår egen metod är det definitivt, den är jättestor.

B: Den är normativ så att säga?

P: Ja. Medvetet så vet man att inga eller väldigt få projekt kommer att behöva allting på bredden och höjden och djupet.

B: Mmmm.. men är det okej att så att säga... "ja nu har vi lagt till det och tagit bort det" och så vidare... "Det här är det vi skall göra", är det okej att inte följa dom här templatens slaviskt eller är det så att: "nu jäklar kör vi så här"?

P: Det är också varierande och det blir väl viktigare ju större och mer internationellt ett projekt är. För om vi har ett projekt här, lokalt i Göteborg, tillsammans med ett företag någonstans på Hisingen...

B: ...en biltillverkare som vi inte nämner vid namn...

M: ...med blåvit logotyp.[skratt, alla]

P: Så.... är det ju ingen... det finns ingen internpolis som granskar projekten och säger att här har ni inte använt metoden på rätt sätt, det finns gott om interpoliser annars som kollar finanser och sådant men så länge kunden är "happy" i det specifika fallet... Och när man gör en sådan anpassning så är det ju tillsammans med kunden... "vi vill göra så här men vi brukar göra på det här sättet", "ja okej då vi brukar göra så här för det gör vi överallt i världen men visst kan vi lämna dom här två sakerna om det passar er". Nej det är inte heligt bara för att det skall vara heligt.

M: På vilken nivå kan man så att säga...hur djupt går den där oheligheten om vi skall kalla den för det? Anta nu att man skall ta ett sådant här beslut att kunden säger att: "ni kan ju göra så men det kanske passar bättre för oss att göra så här den här gången". Vem är det som tar det beslutet, är det...?

P: Det är väldigt lågt i organisationen...

M: Okej, så det kan vara individuella systemintegratörer eller systemutvecklare som tar det beslutet helt enkelt i ett projekt?

P: Ja, precis. Kanske tidigare i projektet att man kommer överens i någon strategifas eller försäljningsfas, att vi kör RUP som huvudmetod fast vi tar inte med dom här delarna och då är det väl typiskt framtida program eller projektledare liksom tittar på helheten. Och längre fram i projektet kan man säkert göra anpassningar på lägre nivå då att: "okej om vi använder Clearcase... vi modifierar det för det finns säkert massa standardfält som man skall fylla i men vi har inget behov av det så vi hoppar över det". Ett typiskt lokalt beslut som programledningen inte känner till, och det går att göra på alla nivåer.

B: Vi är väl egentligen nöjda, tack för att vi fick komma.