

Vidareutveckling i standardsystemkontext

Magisteruppsats, 10 poäng, inom systemvetenskapliga programmet

Framlagd: April 2006

Författare: Natalia Hajas
Jasmina Ramicevic

Handledare: Jonas Hedman

LUNDS UNIVERSITET
Informatik

Vidareutveckling i standardsystemkontext

© Natalia Hajas
Jasmina Ramicevic

Magisteruppsats framlagd april 2006
Omfång: ca 70 sidor
Handledare: Jonas Hedman

Resumé

Standardsystem är något som är mycket vanligt hos företag idag. Det är många som implementerar sådana system för att effektivisera sin verksamhet. När verksamheten hos ett företag förändras eller om verksamheten skiljer sig något från andra uppstår vissa problem med standarder i system. I det läget kan det bli nödvändigt att vidareutveckla ett standardsystem för att det ska passa just det företagets verksamhet. Vidareutveckling kan till exempel ske i form av anpassningar eller i form av förvaltning. Det som ska vidareutvecklas kan upptäckas redan i säljfasen av ett standardsystem eller i ett senare skede då företaget får en förändring i sin verksamhet. Denna typ av utveckling är dock hittills mycket svagt utforskad och det finns inga ramverk för vidareutveckling. Syftet med denna undersökning är att utveckla ett ramverk för vidareutveckling i standardsystemkontext. Detta sker genom en empirisk undersökning med intervjuer och dokumentationsgranskning. Resultatet visas genom ett ramverk där de olika faserna i vidareutveckling av standardsystem presenteras: *inledning, utveckling, implementering* samt *drift och underhåll*. Dessutom visas de påverkande faktorerna: *verktyg, kompetens, användarsynpunkter* och *kritiska framgångsfaktorer*. Även *förändringsbehovet* av vidareutveckling presenteras; Kunden vill ha ett system som passar in på verksamheten. Sist kan man se *resultat*, vilket är den nytta som kunden får ut av vidareutveckling. Vidareutveckling skiljer sig inte mycket från vanlig standardsystemutveckling i processen. De saker som är unika med vidareutveckling är att man i vidareutvecklingen använder utvecklingsverktyg som är mycket effektiva, att man måste ta hänsyn till körande installationer och att utvecklingsprocessen har en kortare inledningsfas, vilket även gör att hela processen blir kortare.

Nyckelord

Vidareutveckling, Systemutveckling, Standardsystem

Förord

Vi önskar här ta tillfället i akt att tacka några personer som har gjort det möjligt för oss att sätta samman den här uppsatsen. Utan dem hade det varit omöjligt.

Vi vill rikta ett stort tack till de fem personer som har ställt upp med sin tid och låtit oss intervjua dem. De fem personerna är: Andreas Quist på Intentia, Olle Andersson och Peter Bjellerup på IBM, Anders Carlsson på q2d Solutions samt Per Kronqvist på Unikum vilka har bidragit med oerhört mycket värdefull information till uppsatsen.

Vi vill även rikta ett stort tack till vår handledare, Jonas Hedman, som har hjälpt oss väldigt mycket med denna uppsats.

2006-04-08

Lund

Natalia Hajas och Jasmina Ramicevic

Innehållsförteckning

1 Inledning	7
1.1 Bakgrund	7
1.2 Problemformulering	7
1.3 Syfte	9
1.4 Avgränsningar	9
1.5 Disposition över uppsatsens kapitel	9
2 Genomförande	11
2.1 Forskningsansats	11
2.2 Litteraturgranskning	11
2.2.1 Källkritik	12
2.3 Datainsamlingsmetod	12
2.3.1 Intervjuer	13
2.3.2 Dokumentation	14
2.3.3 Urval av undersökningsobjekt	14
2.3.4 Utformning av intervjuguide	15
2.4 Etiska aspekter	16
2.5 Analysmetoder	16
2.5.1 Transkribering	16
2.5.2 Kodning och analys	17
2.6 Utformning av ramverk	19
2.7 Validitet och reliabilitet	20
3 Systemutveckling i standardsystemkontext	22
3.1 Systemutveckling	22
3.2 Systemutvecklingsfaser	23
3.3 Systemutvecklingstyper	23
3.3.1 Traditionell systemlivscykel	23
3.3.2 Prototyp	26
3.3.3 Standardsystem	27
3.4 Utmaningar i systemutveckling	28
3.5 Kritiska framgångsfaktorer	28
3.6 Standardsystem som ett informationssystem	29
3.6.1 Nyttan med standardsystem	29
3.6.2 Utmaningar med standardsystem	30
3.7 Vidareutveckling	30
3.7.1 VFS-modellen – en utvecklingsmodell i standardsystemkontext	31
3.7.2 Anpassning	32
3.8 Verktyg	33
3.9 Sammanfattning av teorin	34
4 Presentation av empiriska resultat	36
4.1 Intention	36
4.2 IBM	40
4.3 q2d Solutions	44
4.4 Unikum	47
5 Analys	50
5.1 Vidareutveckling	50
5.1.1 Metod	54
5.2 Verktyg	60
5.3 Dokumentation	61

5.4 Användarinvolvering	62
5.5 Sammanfattning av analys	63
6 Resultat – Ramverk för vidareutveckling i standardsystemkontext	64
6.1 Typer av vidareutveckling	64
6.2 Unikt med vidareutveckling	64
6.3 Problem som kan dyka upp vid vidareutveckling	65
6.4 Ramverk	65
6.4.1 VUS	65
6.5 Förändringsbehov	67
6.6 Metoden	67
6.6.1 Inledningsfas	67
6.6.2 Utveckling	68
6.6.3 Implementering	68
6.6.4 Drift och underhåll	68
6.7 Påverkande faktorer	68
6.7.1 Verktyg	68
6.7.2 Användarinvolvering	69
6.7.3 Kompetens	69
6.7.4 Kritiska framgångsfaktorer	69
6.8 Resultat av vidareutvecklingen	69
6.9 Ramverkets relevans i standardsystemkontext	69
7 Slutsats	71
7.1 Förslag till framtida forskning	72
Referenser	73
Litteratur	73
Avhandlingar och uppsatser	74
Internet	74
Bilagor	75
Intervjuguide 1	75
Intervjuguide 2 – Reviderad	76
Informerat samtycke	77

Tabellförteckning

Tabell 1: En uppställning av intervjupersonerna. Titlarna är skrivna på både svenska och engelska då de varierar på samma sätt i verkligheten. Vi har följt det som finns angivet på respektive visitkort. _____	14
Tabell 2: Exempel på öppen kodning _____	18
Tabell 3: Kategorier med bokstavsbeteckningar _____	19
Tabell 4: Kortfattad beskrivning av processerna i vidareutvecklingen _____	50
Tabell 5: Unikhet i vidareutveckling. En presentation av de saker som skiljer sig när det kommer till vidareutveckling _____	51
Tabell 6: En presentation av de problem som finns kring vidareutveckling _____	52
Tabell 7: Övergripande metoder som används i vidareutveckling _____	54
Tabell 8: Inledningsfasen av vidareutvecklingsprocessen närmare beskrivet _____	55
Tabell 9: Utvecklingsfasen i vidareutvecklingsprocessen _____	55
Tabell 10: Implementeringsfasen i korthet _____	56
Tabell 11: Kortfattad beskrivning av fasen drift och underhåll _____	56
Tabell 12: En presentation av kvalitetssäkring och kritiska framgångsfaktorer kring vidareutveckling _____	57
Tabell 13: En presentation av kompetens och hur arbetsteamerna kan se ut vid vidareutveckling _____	58
Tabell 14: Verktyg som företagen använder vid vidareutveckling _____	60
Tabell 15: Beskrivning av hur dokumentationen ser ut vid vidareutveckling _____	61
Tabell 16: Beskrivning av användarinvolveringen vid vidareutveckling _____	62
Tabell 17: En översikt av analysen _____	63

Bildförteckning

Teckning 1: Utvecklingen med inriktning mot standardsystem och dess faser (Brandt et al. 1998, s 109). _____	32
Teckning 2: Arbetsstegen för anpassning av standardsystem där punkt 11 är inringad eftersom det är det området vi är inne på (Brandt et al. 1998, s 120). _____	33

Illustrationsförteckning

Illustration 1: VUS, VidareUtveckling av Standardsystem _____	66
---	----

1 Inledning

I det här kapitlet redogör vi för bakgrunden till ämnesvalet, problemformuleringen och syftet med undersökningen. Sist visar vi en disposition över uppsatsen.

1.1 Bakgrund

Under vår studietid på Systemvetenskapliga programmet vid Lunds universitet har vi kommit i kontakt med många modeller kring informationssystem, IS. Vi har studerat såväl projektprocesser som utvecklingsmetodik och vilka faktorer som påverkar dessa. Vi har även kommit i kontakt med många typer av IS.

I den här uppsatsen är vi främst inriktade på en speciell gren av informationssystem, nämligen *standardsystem*. Det främsta argumentet till detta var att vi ville fördjupa oss i ämnet informationssystem i allmänhet och i standardsystem i synnerhet. Vi ville bland annat lära oss mer om det för att förbereda oss inför arbetslivet där de flesta företag har någon koppling till den här typen av system. Eftersom vi även är intresserade av projektprocesser ville vi gärna studera standardsystem just ur den synvinkeln. Vi menar alltså att standardsystem i detta kontext ska förstås som ett sätt att utveckla system.

Vi vill även poängtera att målgruppen för den här uppsatsen i första hand är systemvetare. Således har vi utgått ifrån att vissa begrepp inte behöver förklaras närmare eftersom vi menar att man som systemvetare har en viss förkunskap som gör att man förstår uppsatsens innehåll mer naturligt.

1.2 Problemformulering

Under vår förstudie av konceptet standardsystem växte det fram en synvinkel som intresserade oss särskilt mycket. Vi ställde oss bland annat frågorna: Vad händer när ett standardsystem inte passar ett företags verksamhet? Hur gör leverantörerna för att få standardsystemen att passa alla typer av företag? Genom de frågorna väcktes vårt intresse för *vidareutveckling av standardsystem*. Vi insåg dock mycket snabbt att vi inte kunde komma längre i litteratursökningen om vidareutveckling. Det finns helt enkelt inga modeller som rör vidareutveckling av standardsystem specifikt. Det lilla som vi kom fram till via litteraturen var bland andra följande förklaringar av Wiktorin (2003):

- Vidareutveckling ses ofta som en del av ett systems förvaltningsfas.
- En typ av vidareutveckling av informationssystem är *systemintegration* eller *sammankoppling av befintliga system*.

- En annan typ av vidareutveckling är när befintliga system i företagen underhålls och vidareutvecklas och dessa aktiviteter svarar för två tredjedelar av informationssystemens kostnader på ett företag.

Det är många företag idag som implementerar standardsystem för att effektivisera sin verksamhet men när verksamheten hos ett företag förändras eller om verksamheten skiljer sig från andra kan det uppstå vissa problem med standarder. I det läget kan det bli nödvändigt att vidareutveckla ett standardsystem för att det ska passa just det företags verksamhet. Vidareutveckling kan till exempel ske i form av anpassningar eller i form av förvaltning. Det som ska vidareutvecklas kan upptäckas redan i säljfasen av ett standardsystem alternativt i ett senare skede då företaget får en förändring i sin verksamhet varför denne i det fallet kan behöva vidareutveckling av tilläggsapplikationer.

Vi anser att vidareutveckling är viktigt eftersom standardsystem omöjligt kan täcka alla företags verksamheter och önskemål även om systemen är mycket standardiserade. Vissa anpassningar måste i princip förekomma om systemet ska nå en tillräckligt god träffyta, menar Brandt et al. (1998) och Hedman & Kalling (2002). Vid förändringar i företagets verksamheter uppkommer nya informationsbehov som inte kan hanteras på grund av systemets bristande flexibilitet (Andersson et al. 2005). Därför finns det behov av att vidareutveckla, för att därigenom kan man skapa bättre samordning i standardsystem och bättre passform mot företags verksamhet. Med detta i åtanke blev vi intresserade av följande frågor:

1. *Hur ser utvecklingsprocessen ut vid vidareutveckling av standardsystem?*
2. *Vad är unikt med vidareutveckling av standardsystem jämfört med traditionell systemutveckling?*

Dessa frågor har drivit vår undersökning framåt och hjälpt oss att uppnå vårt huvudsakliga syfte med undersökningen som vi presenterar i stycke 1.3. Forskningsfrågorna bör således inte ses som något huvudmål i den här uppsatsen utan bara delmål. Syftet och frågorna är tänkta att hänga ihop på så vis att forskningsfrågorna ska väcka intresse för området och med sina svar leda till att huvudmålet, syftet med undersökningen, kan uppnås.

I litteraturgenomgången har vi kommit över många böcker, avhandlingar och uppsatser inom det vida ämnet standardsystem. Det är många före oss som har intresserat sig för ämnet ur olika aspekter, till exempel: Ekman (2004), Hedman (2003), Kaldren (1995) och många fler, samt ett stort antal uppsatsskribenter på både kandidat och magisternivå. Vi är långt ifrån de första som har valt att göra en studie inom ramen för standardsystem. Vi har dock hittills inte stött på någon publikation som handlar om den specifika inriktningen som vi har intresserat oss för. Vi såg alltså ett problem; Det finns inga ramverk för vidareutveckling i standardsystemkontext. Detta gjorde att det blev, om möjligt, ännu mer intressant att genomföra den tänkta studien.

I detta läge ansåg vi att det bästa sättet att samla in kunskap om hur vidareutvecklingsprocessen ser ut var att gå raka vägen till personer som har erfarenheter inom ämnet. Därför valde vi att genomföra vår studie genom att intervjua ett antal företag och därmed hämta kunskap direkt från de som sysslar med vidareutveckling.

1.3 Syfte

Syftet med denna undersökning är att utveckla ett ramverk för vidareutveckling i standardsystemkontext.

1.4 Avgränsningar

Vår undersökning har avgränsats genom att vi endast har valt att titta på vidareutveckling från leverantörernas perspektiv och inte från kundens. Så som vi beskrev i problemformuleringen ovan så avses här med vidareutveckling såväl anpassning av standardsystem som utveckling av tilläggsapplikationer i både sälj- och förvaltningsfas. Vidare har vi valt att exkludera vissa delar av den teori som normalt ingår i Alters (1999) beskrivning av olika typer av systemutveckling. Det sistnämnda förklaras närmare i kapitel 3.

1.5 Disposition över uppsatsens kapitel

Inledning	Kapitel 1 – Här redogör vi för bakgrunden till ämnesvalet, problemformuleringen och syftet med undersökningen.
Genomförande	Kapitel 2 – Här beskriver vi våra metoder för undersökningen. Vi diskuterar en lämplig forskningsansats, datainsamlingsmetoder och analysmetoder. Vidare redogör vi för undersökningens reliabilitet och validitet.
Systemutveckling i standardsystemkontext	Kapitel 3 – Här förklarar vi några grundläggande begrepp och teorier bakom systemutveckling. Vidare förklarar vi även konceptet standardsystem i stora drag.
Presentation av empiriska resultat	Kapitel 4 – Här presenterar vi all information som vi har inhämtat under den empiriska undersökningen genom berättelser.
Analys	Kapitel 5 – Här kommer uppsatsens empiriska resultat att granskas och analyseras med hjälp av teorin.
Resultat - Ramverk för vidareutveckling i standardsystemkontext	Kapitel 6 – Här presenterar vi ett ramverk som vi har identifierat utifrån våra tolkningar av de empiriska resultaten samt teorin.
Slutsats	Kapitel 7 – Här presenterar vi våra slutsatser där vi återknyter till våra inledande frågeställningar och syfte med undersökning.

2 Genomförande

I det här kapitlet beskriver vi våra metoder för undersökningen. Vi diskuterar en lämplig forskningsansats, datainsamlingsmetoder och analysmetoder. Vidare redogör vi för undersökningens reliabilitet och validitet.

2.1 Forskningsansats

En vetenskaplig undersökning kan till exempel göras empiriskt, det vill säga genom observationer av verkligheten. Genom att skaffa erfarenheter av omvärlden kan den göras förståelig (Patel & Davidsson, 1994). Med den utgångspunkten i åtanke anser vi att undersökningen i uppsatsen är en *empirisk undersökning*. Eftersom vi har samlat in mycket av informationen genom intervjuer har vi vänt oss till verkligheten för att söka svar på våra frågeställningar. Då det har varit svårt att finna litteratur som förklarar vidareutveckling av standardsystem blev det nödvändigt att göra en empirisk undersökning genom att uppsöka företag som sysslar med den typen av utveckling och göra intervjuer. Samtidigt såg vi även möjligheten att kunna bidra med ny kunskap om systemutveckling genom den insamlade informationen.

Vi har även valt att inta en kvalitativ forskningsansats i den här studien. Kvalitativ forskning handlar om att djupare kunna beskriva företeelser ur en individs synvinkel, säger Bryman (1997). Det ska vara den studerade individen, i vårt fall individerna på de företag som vi har intervjuat, som för undersökningen framåt snarare än att denne styrs av forskaren, menar Bryman. Genom att vi koncentrerade oss på några enstaka företag istället för många menar vi att vi kunde skaffa oss insikter som kan få vidare konsekvenser och enligt Denscombe (2000) skulle de insikterna inte ha sett dagen om vi hade valt att använda en undersökningsstrategi som försöker täcka ett stort antal enheter. Eftersom en kvalitativ undersökningform kunde erbjuda fler möjligheter att gå tillräckligt mycket på djupet för att reda ut komplexiteten i vår frågeställning så valde vi den framför en kvantitativ undersökningsstrategi.

2.2 Litteraturgranskning

Vi har valt att inhämta stora delar av kunskapen om standardsystem och systemutvecklingsprocesser via böcker, uppsatser och tidskrifter. För att kunna bidra med kunskap till vår undersökning var vi tvungna att förse oss själva med goda förkunskaper om systemutveckling och om standardsystem. Därefter kunde vi genomföra vår studie om vidareutveckling av standardsystem. Med den förkunskap vi samlade in genom den befintliga informationen kunde vi ställa mer relevanta frågor under våra intervjuer för att få ytterligare upplysningar om vår specifika studie. På det viset fick vi även bättre möjligheter att kunna koppla vårt

informationsfynd i undersökningen till teorierna som finns och därmed se eventuella samstämmigheter eller olikheter, vilka var mycket värdefulla för våra diskussioner. En koppling mellan det teoretiska och det empiriska planet är nödvändig eftersom utan en sådan koppling är den vetenskapliga undersökningen meningslös (Svenning 1999). Denna koppling bidrar således även till validiteten i vår undersökning.

Vi vill här göra läsaren uppmärksam på att en presentation av information alltid föregås med en viss reducering (Miles & Huberman, 1994). Litteraturgenomgången i den här undersökningen bör därför ses som mycket översiktlig och innefattar främst det nödvändiga för att vi ska kunna ta fram ett ramverk för vidareutveckling i standardsystemkontext. Vi har även presenterat lite generella teorier om standardsystem för att underlätta för läsaren att hänga med i ämnet.

2.2.1 Källkritik

Det har för oss varit praktiskt omöjligt att ta oss igenom all relevant litteratur under den korta period som var avsedd för undersökningen. Därför har det varit svårt att avgöra hur bra kvalitén är i den litteratur som finns. Av den anledningen har vi tagit lite hjälp av vår omgivning till detta. Vårt urval av litteratur har därmed främst baserats på rekommendationer från vår handledare och från vänner och kurskamrater som tidigare har skrivit uppsatser om systemutveckling. Dessutom har vi till viss del sökt upp litteratur genom att kontrollera referenslistan hos tidigare uppsatser som har liknande tema och sist men inte minst har vi även dragit nytta av litteraturerfarenheter från tidigare kurser.

När det kommer till Internet så har vi besökt hemsidorna till de företag som vi har träffat och intervjuat. Det gällde då framförallt informationssökning angående adress samt för att förstudera företaget innan intervjutillfället. Vi har även besökt hemsidor till verktygsutvecklare för att söka information om några av de verktyg som används vid vidareutveckling. Vi anser att de sidor som vi har besökt är mycket tillförlitliga eftersom det är företagens officiella hemsidor.

Vi vill här också påpeka att även om vår studie undersöker leverantörens perspektiv, så hittade vi i *VFS-modellen* (Brandt et al., 1998), som beskrivs närmare i stycke 3.6.1, några riktlinjer som var hjälpsamma i vårt försök att närma oss vidareutvecklingen av standardsystem. Det är anledningen till att vi har tagit upp den modellen i litteraturgranskningen. Vi ger dock inte någon djupare förklaring av modellen, utan presenterar den kortfattat med fokus på anpassning.

2.3 Datainsamlingsmetod

Enligt Yin (2003) är en empirisk undersökning beroende av mer än en beviskälla för att kunna säkra kvalitén i den. Vi har gjort datainsamlingen genom två kanaler: intervjuer med experter på fältet och dokumentationsgranskning.

Dokumentationen som vi har granskat kommer från de företag som vi har intervjuat och är således en del av det empiriskt insamlade materialet.

2.3.1 Intervjuer

Som en första kanal för datainsamling gjorde vi intervjuer. Närmare bestämt genomförde vi intervjuer med fyra olika företag som sysslar med vidareutveckling av affärssystem i någon form. Detta hjälpte oss att göra en jämförelse mellan metoderna som de olika företagen använder vid vidareutveckling och traditionella systemutvecklingsmetoder. Detta i sin tur hjälpte oss att uppnå vårt syfte med undersökningen, nämligen att hitta ett ramverk för vidareutveckling i standardsystemkontext.

När det kommer till val av plats för intervjuerna så har vi valt att göra intervjuerna ute på fältet trots Denscombes (2000) resonemang om att forskaren har bättre kontroll över intervjun i egen miljö. Vår personliga uppfattning var att det var lättare att uppnå att intervjupersonerna kände sig trygga i situationen om vi befann oss på deras hemmaplan. Dessutom ansåg vi att det även var lättare att få våra intervjupersoner att vara engagerade om intervjun inte behöver innebära besvär i form av förflyttningar för dessa.

Kvale (1997) anser att intervjuare bör ha förberett sig noga inför en intervju. Man bör skaffa sig förkunskap om ämnet, formulera ett klart syfte med intervjun och känna till olika intervjutekniker och besluta sig för en. Vi förberedde oss genom att läsa på om systemutveckling och standardsystem i litteraturen. Därefter formulerade vi syftet att försöka utveckla ett ramverk för vidareutveckling i standardsystemkontext, som vi presenterade för de personerna som vi kontaktade för intervjuerna. Vi beslutade oss för att göra semistrukturerade intervjuer där vi tog stöd av en intervjuguide (bilaga 9.1 och 9.2) och dessutom spelade vi in intervjuerna med en MiniDisc-spelare (MD-spelare) vilket hjälpte oss att ta tillvara allt som diskuterades under intervjun. Eftersom vi inte behövde koncentrera oss på att hinna anteckna allt viktigt som sades kunde vi vara avslappnade och deltaga i diskussionen på ett mer naturligt sätt. Vi upplyste dessutom våra intervjupersoner redan vid förfrågan om intervjun om hur lång tid vi uppskattade att intervjun skulle ta. Vi räknade med ungefär en timme, plus minus en kvart för hela intervjun. Detta höll vi oss inom på alla utom en, på vilken det deltog två intervjupersoner och tiden uppgick till en och en halv timme.

Intervjuerna var, som sagt, av semistrukturerad karaktär, vilket innebär att de delvis var styrda med intervjufrågor men lämnade samtidigt mycket utrymme för diskussioner. Den intervjuade personen hade möjlighet att tala fritt och det fanns också möjlighet för oss att ställa följdfrågor som inte var formulerade från en mall. Vi anser dock att det var nödvändigt att ha en intervjuguide att utgå från eftersom det gjorde att samtalet inte spårade ur för mycket och gled iväg åt andra teman som faller utanför vår undersökning.

2.3.2 Dokumentation

Vid våra besök hos företagen som vi har intervjuat har vi erhållit en del dokumentation i form av broschyrer över företagsverksamheten samt PowerPoint-presentationer och Worddokument rörande utvecklingsmetoder. Detta material kan vi inte presentera i sin helhet med anledning av att vi garanterat företagen konfidentialitet. Materialet har dock varit till stor nytta när vi behövde klargörelse i vissa delar av utvecklingsmetoderna.

2.3.3 Urval av undersökningsobjekt

Det avgörande kriteriet för valet av vilka företag som innefattades av undersökningen var deras verksamhet. Efter en förundersökning av vilka företag som har den rätta verksamheten kontaktade vi de vi fann, vilket var runt 25 stycken, och fick med lite tur samarbetsvilja hos fyra företag, q2d, Intenia, Unikum, och IBM. Dessa fyra rymdes inom ramen för rätt verksamhet vilket är *vidareutveckling av standardsystem*. Även om detta inte är alla fyras huvudverksamhet så sysslar alla med det i någon grad. Eftersom vårt urval främst baserades på teoretiska grunder och inte slumpmässiga, det vill säga, vi har valt fall efter vissa kvaliteter hos undersökningsobjekten, så kan vårt urval anses vara ett så kallat *selektivt urval* (Svenning, 2003). Dessa fyra företag tillsammans anser vi representerar en god vidd inom området. Unikum och Intenia sysslar med utveckling av standardsystem i huvudsak men de är av mycket olika storlek. q2d sysslar enbart med vidareutveckling av standardsystem. IBM sysslar inte med utveckling av standardsystem men de samarbetar med till exempel både Intenia och SAP vilkas produkter de vidareutvecklar. Tack vare detta tror vi att vi har fått ett relevant kunskapsbidrag i vår empiriska undersökning.

De personer som har intervjuats har något olika bakgrunder och positioner på sina företag. Det som är gemensamt är att samtliga representerar företagen med en helhetsöverblick över verksamheten, vilket gör att de är lämpliga för vår undersökning. När vi kontaktade företagen försökte vi få tag i just sådana personer i företaget som har bra överblick av hela verksamheten. De personer vi har intervjuat är följande:

Företag	Namn och titel på intervjuperson
Intenia	Andreas Quist – Human Resource Manager, Nordic Services
IBM	Olle Andersson – Managing Consultant/Business Manager Peter Bjellerup – Usability Consultant
q2d Solutions	Anders Carlsson – Senior Consultant/Managing Director
Unikum	Per Kronqvist – Seniorkonsult

Tabell 1: En uppställning av intervjupersonerna. Titlarna är skrivna på både svenska och engelska då de varierar på samma sätt i verkligheten. Vi har följt det som finns angivet på respektive visitkort.

Vårt initiala mål var att genomföra 10 – 15 intervjuer men vi insåg senare att detta inte skulle bli nödvändigt eftersom vår undersökning är av kvalitativ karaktär. Svenning (2003) säger att eftersom en kvalitativ undersökning snarare syftar till att exemplifiera än att generalisera så räcker det med att välja ut 4-5 stycken för intervjuer. Eftersom vi just gör en kvalitativ undersökning och vårt syfte inte är att generalisera utan att exemplifiera så anser vi att 4 olika företag med totalt fem intervjuade personer bör vara tillräckligt för vår undersökning.

2.3.4 Utformning av intervjuguide

Efter att ha försett oss själva med förkunskaper om ämnet kunde vi formulera mer relevanta och intressanta intervjufrågor som bidrog med riklig information till vår studie. Urvalet av frågorna som vi har använt på intervjuguiden har baserats på teorier om bland andra *traditionell systemlivscykel, standardsystem samt prototyputveckling* (Alter, 1999). Dessa har delats upp i ytterligare kategorier *inledning, utveckling, implementering* samt *drift och underhåll*. Vidare, inom dessa kategorier förklaras dokumentation, problematik kring systemutveckling och så vidare. Vi har ställt frågorna utifrån denna litteraturuppställning. Frågorna är ämnade att försöka ta fram de skillnader som eventuellt föreligger mellan traditionell systemutveckling och vidareutveckling. På det viset menar vi att vi senare ska kunna urskilja ett ramverk för vidareutvecklingsprocessen.

Från början formulerade vi ett antal frågor som vi ansåg skulle täcka det informationsbehov vi var ute efter att täcka. Dessa frågor gjorde vi först ett pilottest på varefter vi konsulterade med vår handledare om vi kunde använda frågorna. Därefter använde vi dem till första intervjun. Det visade sig dock under den första intervjun att några av frågorna var överflödiga varpå vi formulerade om intervjuguiden och kortade ner den. Den nya versionen använde vi sedan till resterande intervjuer. Skillnaden mellan de två intervjuguiderna går att se tydligt i bilaga 9.1 och 9.2 varför vi inte exemplifierar det här.

För att öka möjligheten till att frågorna blev korrekta och välformulerade så har vi studerat intervjufrågorna på en försöksperson innan den första verkliga intervjun. Denna person berördes inte av undersökningen på något vis utan var helt utomstående. Detta gav dock inte så mycket just av den anledningen att personen inte var insatt i ämnet. De saker som var oklara för denna var sådant som vad standardsystem är till exempel. Vi insåg att vi skulle bli tvungna att prata med någon som hade kunskap i ämnet. I brist av tid inför den första intervjun fick det bli vår handledare direkt som fick granska frågorna. Han ansåg att vi skulle prova de frågorna vi hade i den första intervju och låta den intervjun vara ett prov i sig. Det som förändrades i intervjuguiden finns att se i bilaga 9.1 och 9.2 som vi nämner ovan. Anledningen till förändringarna var att vi insåg att några av frågorna var lite klumpigt formulerade och vissa tyckte vi i efterhand att de egentligen inte var så relevanta för vår undersökning varför de blev överflödiga.

2.4 Etiska aspekter

De etiska aspekterna utgör en viktig fråga i en undersökning, anser vi, varför vi har tagit hänsyn till dessa i vår studie. Vi har garanterat intervjupersonerna att ingen information som är konfidentiell kommer att visas upp. Det är intervjupersonen och det företag denne representerar som avgör vilken information vi kan ta med i vår undersökning. Konfidentialitet syftar till överenskommelser med undersökningspersonerna om vad som får och inte får göras med informationen man har samlat in (Miles & Huberman, 1994).

Vi har lämnat information om *informerat samtycke* till våra intervjupersoner och samtliga har skrivit på detta. Informerat samtycke-dokumentet finns bifogade som bilaga 9.3. Vi har därmed respekterat konfidentialiteten genom att vi låtit intervjupersonerna läsa igenom råmaterialet i form av transkriberingar, berättelserna samt den slutliga uppsatsen innan den gick till publicering. Därmed gav vi dem en möjligheten att ta tillbaka information som de inte ville gå ut med offentligt.

2.5 Analysmetoder

Som analysmetoder har vi valt att göra transkriberingar av inspelningarna för att vidare kunna koda och tolka det. Därefter var det möjligt att hitta kategorier och mönster som senare gjorde att vi kunde analysera och diskutera materialet.

2.5.1 Transkribering

Eftersom vi har valt att göra intervjuer som en datainsamlingsmetod var vi även tvungna att transkribera inspelningarna, det vill säga skriva ut inspelningarna till text. Därefter blev det möjligt att enklare analysera materialet.

Vid transkriberingen delade vi upp ljudfilen i två delar och arbetade med varsin del på varsitt håll. Detta gjorde vi med alla fyra intervjuer. Vi märkte snabbt att det ibland fanns vissa bekymmer med att höra vad intervjupersonerna har sagt varför vi markerade dessa partier med färg. När vi sedan läste varandras delar kunde vi återigen lyssna för att se om den andra kunde höra istället. Nedan visar vi ett utdrag ur en av transkriberingarna som exempel.

J: 13.10 Okej. När ni får den här kravspecifikation börjar ni med att börja skissa? Börjar ni med att sätta på skärmen nånting?

A: Nej...

J: Eller diskuterar ni?

A: Vi försöker diskutera, ja. Så vi har en kravspec nu inifrån Danmark som vi höll på med rätt så mycket. Det är sån där gamla skolan med 50 sidor någonting och så beskrivet ner på fältnivå vad de vill ha. Det är hopplöst när man inte får en övergripande förklaringar och när man liksom inte kan prata med dem om vad de egentligen vill ha. Sen har vi i portaler... finns det dessutom produkter som man...så att säga program som bara snorar i

portalen som inte behöver gå in i backend system. Bland annat ett sånt där projektplaneringsprogram som suger information från något som heter projektsystem inne i SAP och finanssidan och sedan kör man alltihop inne i portalen så det är funktionaliteten här också. Det måste man komma ut och visa för de och liksom förklara att...upplysa kunden att det här finns och såna saker... Så nu ska vi egentligen dit och visa dem. Jag fattar inte varför vi höll på med dessa papper tidigare, så det är helt bortkastat. Så det ska ligga på ett mycket högre nivå. Det ska vara såna här dataflöden eller annat som liksom visar processer. Text är svår att förstå. Man ser inte sammanhang.

N: Gör ni fältstudie hos företag som ni ska utveckla hos...händer det ofta att ni går ut och observerar hur de fungerar på sina arbetsplatser?

A: Ja, vi gör inte det, men det är någonting vi liksom strävar efter att få göra.

2.5.2 Kodning och analys

Det finns några olika huvudtekniker att koda en transkriberad text med. Miles & Huberman (1994) föreslår till exempel berättelser, tolkningar och kodning efter mönster och kategorier. Kvale (1997) föreslår liknade metoder samt meningskoncentrering, meningskategorisering och ad hoc. Ad hoc är den teknik vi främst har valt att följa eftersom den uppmanar till att använda olika tekniker fritt och växla mellan dessa. De tekniker vi har växlat mellan är just kategorisering och meningskoncentrering. Vi har skrivit berättelser för varje enskilt företag för att beskriva de fyra intervjuerna lite djupare. Vi har även gjort tolkningar av fakta i materialet när vi har diskuterat och analyserat resultatet.

För att säkra kvalitén på resultatet ytterligare har vi först genomfört kodningen på var sitt håll. Vi har kodat råmaterialet på två sätt. Först har vi gjort en variant av referenskodning i transkriberingen, vilket vi gjorde under tiden vi transkriberade. Med detta menas att vi har skrivit ut tidpunkter i texterna vid varje tillfälle någon av oss ställer nya huvudfrågor till informanterna. Detta gjorde vi för att på ett enkelt sätt kunna hitta i ljudfilen igen om vi behövde gå tillbaka och säkra att det som står i transkriberingen verkligen var det som sades. Detta behövde vi dock aldrig utnyttja eftersom vi inte stötte på sådana funderingar vid läsningen.

Vi gjorde även en så kallad *öppen kodning* (Denscombes, 2000) där vi skrev *lösa kategorier* i marginalen. De lösa kategorierna uppstod i den öppna kodningsprocessen tack vare att vi i tidigare läsning av teorier kunde urskilja liknande indelning. Detta syns redan i vår utformning av intervjufrågorna som vi beskrev i stycke 2.3.4. Nedan visas ett utdrag ur en transkribering. I marginalen kan man se de öppna kategorierna som vi har kodat utefter i början av kodningsprocessen. I detta exempel bör man inte leta efter sammanhang eftersom texterna är lite brutna och förkortade.

Och det kan man se som så att om man påbörjar fel den PCDn som vi alla har gjort för det blir ju så med de här portalerna och så försöker man ju så klart lägga upp någonting och testa och få fram någonting på skärmen och så bygger men vidare på den. Om man haft en taskig lösning från början och inte tänkt igenom det ordentligt så är man rätt så körd faktiskt. Och då kan man tänka sig stort företag så blir ju det katastrof, ett megaarbete att bygga om. ...	Problem (kravspecifikation)
Så det de här viktiga hållpunkterna som är portalspecifikt då. Annars är faserna likartade med förstudien och så vidare.	Processen (faser)
Det är hopplöst när man inte får en övergripande förklaringar och när man liksom inte kan prata med dem om vad de egentligen vill ha.	Problem (kravspecifikation)
Sen har vi i portaler... finns det dessutom produkter som man...så att säga program som bara snurrar i portalen som inte behöver gå in i <i>backend</i> system. Bland annat ett sånt där projektplaneringsprogram som suger information från något som heter projektsystem inne i SAP och finanssidan och sedan kör man alltihop inne i portalen så det är funktionaliteten här också	Portaler
Så det ska ligga på ett mycket högre nivå. Det ska vara såna här dataflöden eller annat som liksom visar processer. Text är svår att förstå. Man ser inte sammanhang.	Kravspecifikation (processer)
N: Gör ni fältstudie hos företag som ni ska utveckla hos...händer det ofta att ni går ut och observerar hur de fungerar på sina arbetsplatser? A: Ja, vi gör inte det, men det är någonting vi liksom strävar efter att få göra.	Användare (kontakt)

Tabell 2: Exempel på öppen kodning

Vi behöll dock inte de kategorierna utan vid den djupare analysen utförde vi en ny kodning och gjorde en mer specifik kategorisering. Den nya kodningen gjorde vi genom att först titta igenom alla de lösa kategorierna. Därefter samlade vi alla stycken eller meningar, som vi hade satt kategorier på, i en sammanställning där vi sorterade in alla meningar och stycken under de lösa kategorierna. Detta gjorde vi också enskilt. Det resulterade i något olika uppdelningar och lite varierande namn på kategorierna. Sedan tittade vi på våra sammanställningar tillsammans och sorterade ut vilka saker vi hade skrivit samma kategorier på. De kategorier och underkategorier som vi var helt överens om satte vi som *funna kategorier*. Dessa fick varsin bokstavsbezeichnung. Se tabell 3 nedan.

Kategori	Bokstavsbezeichnung
Vidareutveckling	VU
Metod	M
Inledning	In
Utveckling	U
Implementering	Im
Drift och underhåll	DoU
Verktyg	V
Dokumentation	D
Användarinvolvering	A
Problem	P
Kvalitetssäkring och kritiska framgångsfaktorer	K
Kompetens och arbetsteam	KoA
Unikhet i vidareutveckling	UVU

Tabell 3: Kategorier med bokstavsbezeichnungar

Vidare gjorde vi så att vi tillsammans kodade hela sammanställningen efter de kategorierna vi var överens om genom att använda bokstäverna. Vi placerade in alla meningar och stycken på den kategori vi båda tyckte att de hörde hemma på. Faktum var att vi hade hittat ungefär samma saker i transkriberingarna båda två så det var inte särskilt svårt för oss att komma överens. Det enda som skiljde var att vi hade funnit olika mycket saker som vi hade markerat. Även de sakerna tog vi med genom att koda med bokstäverna. Detta gjorde att vi tillsammans förhoppningsvis har lyckats täcka det mesta som är viktigt i råmaterialet. Resterande arbete bestod av att skapa tabeller för de olika kategorierna och sätta in all information som vi hade hittat. Dessa tabeller finns i kapitel 5 tillsammans med analysen. Efter detta skrev vi även berättelser för varje enskilt företag som strukturerades med hjälp av ovanstående kategorier. Dessa presenteras i kapitel 4.

2.6 Utformning av ramverk

Vi vill börja med att här visa en definition av ramverk, vilket är vad vi utgår från när vi pratar om ramverk i vår undersökning.

A framework is an extensible structure for describing a set of concepts, methods, technologies, and cultural changes necessary for a complete product design and manufacturing process. (Internet: Cern EDMS, Framework, 20060408)

När vi utformade ramverket som presenteras i kapitel 6 utgick vi huvudsakligen ifrån de empiriska resultaten. Vi ville visa hur vidareutveckling ser ut i praktiken varför vi valde att fokusera på den empiriskt insamlade informationen i första hand. I grunden har vi dragit paralleller till den teori som vi har granskat eftersom den i sig har påverkat vår strategi för den empiriska undersökningen från början.

Ramverket utvecklades således genom att vi jämförde teorin med empirin och det resultat som vi kom fram till var några likheter och några skillnader mellan vanlig systemutveckling och vidareutveckling. Dessa likheter och skillnader utformade vi till kategorier under kodningen och de växte senare fram till byggstenar för ramverket. Vi fick fram en process med fyra faser likt den vanliga systemutvecklingen, dock med några små skillnader. Vidare fick vi fram några faktorer som vi tolkade som att de påverkar processen och dess resultat, vilka skiljde sig lite mer från teorin. Dessa faktorer såg vi inte som speciella faser utan de ingick mer eller mindre i hela processen, varför vi valde att placera dem utanför processen dock i anknytning till den. På det viset växte en bild fram över ett potentiellt ramverk som vi sedan finslipade visuellt. Vi adderade även förändringsbehovsfaktorn i ramverket som en initiering till processen samt resultatet av processen som en utgång ur ramverket eftersom det är där man har uppnått processens målgång. För djupare beskrivning av ramverket hänvisar vi till kapitel 6 där ramverket visualiseras och förklaras närmare.

2.7 Validitet och reliabilitet

Enligt Svenning (1999) går reliabilitet hand i hand med kvantitativa undersökningsmetoder och validitet med kvalitativa. Eftersom vår undersökning är av kvalitativ karaktär ligger den största fokusen på att säkerställa validiteten. Vid diskussionen om validitet brukar man även tala om den *yttre* och *inre* validiteten (Bryman, 1997), där den yttre validiteten betonar generaliserbarheten hos undersökningen och den inre validiteten handlar om hur väl kopplingen mellan teori och empiri är gjord. Den inre validiteten i vår studie kunde vi säkra genom att göra en koppling mellan teorin och våra resultat från den empiriska undersökningen. Detta utförde vi dels genom att presentera våra resultat noga samt genom att presentera befintlig teori om traditionell systemutveckling. I kapitel 5 har vi sedan jämfört och analyserat våra fynd i den empiriska undersökningen med den teori som vi har presenterat i uppsatsen.

Bryman (1997) diskuterar att man kan generalisera kvalitativa studier till teorier, istället för att generalisera själva studien, vilket då skulle kunna motsvara de krav som ställs på generaliserbarheten. Vårt syfte är inte att försöka generalisera undersökningen utan att samla in kunskap och därigenom utveckla ett ramverk som är tänkt att presentera för läsaren en bild av hur vidareutveckling kan se ut i praktiken.

När det kommer till frågan om reliabilitet finns det inte mycket att diskutera eftersom studien är av kvalitativ art, men den faktorn som bidrar till en högre reliabilitet i vår studie är presentationen av vårt tillvägagångssätt i studien. Det

som är intressant i frågan om reliabilitet är huruvida undersökningen är reproducerbar, samt om vi har uppmätt det vi avsåg att mäta (Bryman, 1997). Genom att vi har presenterat och beskrivit våra perspektiv, kontexter, informanter och exakta tekniker utförligt, bör en utomstående åskådare kunna få en relativt god bild av hur vi har arbetat genom undersökningsprocessen och därför bör de kunna ha möjlighet att göra om undersökningen på exakt samma vis under samma omständigheter. Därigenom är det rimligen även möjligt att avgöra om vi har uppmätt det vi avsåg att mäta.

Vi anser även att presentationen av vårt tillvägagångssätt bidrar till att vår studie kan anses vara objektiv, i den mån det är möjligt. Eftersom betraktaren själv kan se hur vi har gått tillväga för att nå de resultat vi har nått kan denne själv avgöra om det vi har kommit fram till är logiskt, även om vi till viss grad kan ha påverkat resultaten genom våra subjektiva tolkningar.

Stärkning av validiteten gjordes även genom att vi skickade våra tolkningar av de intervjuade personers svar tillbaka till dem, så att de skulle bekräfta eller avvisa våra tolkningar samt stryka information som de tyckte var för känslig.

3 Systemutveckling i standardsystemkontext

Eftersom syftet med uppsatsen är att utveckla ett ramverk för vidareutveckling i standardsystemkontext måste vi utgå från vanlig systemutveckling. I det här kapitlet förklarar vi några grundläggande begrepp och teorier bakom systemutveckling. Vidare förklarar vi konceptet standardsystem, anledningar till att företagen skaffar standardsystem samt teorier om kritiska framgångsfaktorer.

3.1 Systemutveckling

Informationssystem har definierats på flera sätt. Verrijn-Stuarts (1989, citerad i Fitzgerald et al., 2002) definierar informationssystem utifrån två perspektiv.

Utifrån ett bredare perspektiv är informationssystem:

...the totality of all formal and informal data representation and processing activities within an organization, including the associated communication, both internally and with the outside world. (Verrijn-Stuarts, 1989, citerad i Fitzgerald et al., 2002, s. 4)

Informationssystem utifrån ett närmare perspektiv definierar han som:

...computer-based subsystems, intended to provide recording and supporting services for organisational operation and management. (Verrijn-Stuarts, 1989, citerad i Fitzgerald et al., 2002, s. 4)

Informationssystemsutvecklingsmetod kan definieras som:

A coherent and systematic approach, based on a particular philosophy of systems development, which will guide developers on what steps to take, how these steps should be performed and why these steps are important in development of an information system. (Fitzgerald et al., 2002, s. 2)

Medan metoden svarar på frågan "what to do", svarar tekniken på frågan "how to do it". Metoden kan bestå av flera tekniker och vilka tekniker som passar in och används bestäms av metoden. (Fitzgerald et al., 2002)

Systemutveckling är en samling av olika aktiviteter som förändrar en organisation genom användning av IT. Systemutveckling innefattar anpassning av befintliga och utveckling av nya informationssystem. (Andersen, et. al., 1994)

3.2 Systemutvecklingsfaser

Varje systemutveckling har fyra faser, oavsett om det är systemutveckling från grunden eller om systemet köps in från en leverantör (Alter, 1999). De faserna är:

- *Inledning* – I inledningsskedet definieras förändringsbehov i befintliga system, intressenter, beslutsfattare samt hur det nya systemet förväntas fungera. Efter att genomförbarhet av förändringar har fastställts, avslutas skedet med en överenskommelse och förslag om hur dessa förändringar ska utföras.
- *Utveckling* – I utvecklingsskedet skaffas och konfigureras hårdvaru- och mjukvarudelar av systemet. Om hårdvaran inte redan finns, köps den och installeras och mjukvaran antingen köps, utvecklas eller anpassas till nya och förändrade behov. I det här skedet skaffas även resurser som icke är IT-relaterade. Fasen inbegriper dokumentation om hur systemet ska fungera och avslutas med olika tester av systemet för att säkerställa att det fungerar som det ska och för att rätta eventuella fel. Även om det fungerar på datorer så betyder det inte att det uppfyller sitt syfte i organisationen.
- *Implementering* – I implementeringsskedet görs systemet fungerande i organisationen. Skedet inkluderar olika aktiviteter som planering, utbildning, konvertering från det gamla till det nya systemet och uppföljning. Planering av konvertering är viktigt, eftersom dåligt utförd konvertering kan orsaka stora oväntade problem.
- *Drift och underhåll* – Detta sker löpande och inkluderar bland annat kontroller för att se att allting fungerar som det ska, rättning av eventuella fel och förbättringar av systemet.

3.3 Systemutvecklingstyper

Det finns fyra alternativa sätt att utveckla informationssystem och de är enligt Alter (1999) följande:

- Traditionell systemlivscykel
- Prototyp
- Standardsystem
- Användarutveckling

Den sistnämnda har vi dock inte gått in närmare på eftersom vi inte anser att det är relevant för vår undersökning.

3.3.1 Traditionell systemlivscykel

Olika författare skriver olika om vilka faser traditionell systemlivscykel består av, men de är eniga i det att varje traditionell systemlivscykel inkluderar förutbestämda steg och aktiviteter. Här kontrollerar man varje steg i

systemutvecklingen med syftet att fastställa att systemet uppfyller alla krav. Nedan ges en kort beskrivning av de olika faserna enligt Alter (1999).

Inledning

Inledningsfasen kan börja på olika sätt, allt ifrån att man i en organisation ser en möjlighet att genom ett nytt informationssystem förbättra verksamheten till att ett befintligt system kraschar och det enda man kan göra är att laga eller byta ut det genom ett större projekt. Oavsett hur fasen börjar består den av två delar eller steg: genomförbarhetsstudie och projektplanering. (Alter, 1999)

Genomförbarhetsstudie brukar ha tre aspekter: ekonomisk, teknisk och organisatorisk. Den är användarorienterad och ifall informationssystem är möjligt att införa produceras en funktionsspecifikation. I specifikationen förklaras problemet, presenteras befintliga affärsprocesser och hur dessa förändras samt ges en uppskattning av nytta, risker och kostnader. Projektplanen visar projektets olika steg med starttiderna, sluttiderna, personal, resurser samt beroenden mellan de olika stegen. (Alter, 1999)

Utveckling

I utvecklingsfasen översätts funktionsspecifikationen till mjukvaruprogram och till hårdvaran. Programmen designas och utvecklas och de olika stegen i fasen säkerställer att systemet uppfyller alla krav. De stegen är enligt Alter (1999):

- *Kravanalys* – Först gör man en kravanalys tillsammans med användarna. Det är en användarorienterad förklaring av vad systemet skall göra. Resultatet blir en *extern specifikation*, vilket är ett dokument där datainput presenteras och det som användarna kommer att se visas. Här går man inte in på detaljer om hur datorn kommer att utföra de olika uppgifterna. Användarna godkänner extern specifikation i fall de anser att förslagen motsvarar deras behov.
- *Intern systemdesign* – Här bestämmer teknisk personal hur processkonfigurering på dator ska göras. Steget resulterar i en intern specifikation. Intern specifikation är en teknisk karta över informationssystemet som visar datormiljön för systemet, detaljerad struktur och innehåll av databasen samt input och output av alla program. Specifikationen jämförs med en extern specifikation och om de stämmer överens godkänns den av teknisk personal.
- *Hårdvaruanskaffning och installation* – Om inte befintliga hårdvarukomponenter kan användas för det nya systemet måste ny hårdvara skaffas och installeras. Man måste ta hänsyn till att de nya komponenterna är kompatibla med de befintliga och man gör en grundlig analys av kompatibilitet, kostnader och kundservice. Användarna är sällan involverade i detta steg.
- *Programmering och enhetstest* – I programmeringssteget skapas datakod som utför uppgifter som beräkningar, datainsamling och rapportgenerering. Fasen går oftast parallellt med hårdvaruanskaffning och installation och inkluderar kodning, test och dokumentation av varje program. Test kallas för *enhetstest*, eftersom här testas man varje programenhet för sig.

- *Dokumentation* – Dokumentation kan göras parallellt med hårdvarusteget och programmeringen. Den produceras av redan befintligt material. För användardokumentation används funktionsspecifikation och externspecifikation. Användardokumentation behöver oftast bli utökad eftersom olika användare behöver veta olika mycket, beroende på deras roller. Internspecifikation och programdokumentation används som bas för dokumentation för programmerare.
- *Systemtest* – Systemtest eller integrationstest kräver att hela systemet testas för att kontrollera att alla delar fungerar som de ska och att de är integrerade för att uppfylla önskade funktioner. Systemtestet brukar avslöja fel och brister i systemet och även i internspecifikationen. Därför måste man gå tillbaka, rätta felen och testa om. En testplan är viktig att ha innan man börjar testa. I den presenteras testdata samt exakt hur testet ska genomföras.

Implementering

Implementeringsfasen börjar efter att man har åstadkommit ett fungerande set av dataprogram och dokumentation. Först gör man en implementeringsplan. Man planerar utbildning, konvertering och godtagandetest.

Med utbildningen menas att lära användarna de nya rutinerna samt hur de ska använda det nya systemet. Hur utbildningen ser ut beror på vilken bakgrund användarna har.

Konvertering från det gamla till det nya systemet måste planeras noga. Man måste vara säker på att datan finns i rätt format och är korrekt. (Alter, 1999) Konverteringen kan göras på flera olika sätt:

- Bestämna tidpunkten och stänga av det gamla samt sätta på det nya systemet med risken att det nya systemet inte fungerar som det ska.
- Konvertera i faser där man inför det nya systemet i delar av verksamheten och behåller det gamla i andra delar. Om något går fel kan man gå tillbaka till det gamla systemet.
- Parallell körning av både det gamla och det nya systemet tills man verifierar att det nya systemet fungerar som det ska.
- En pilotimplementering är oftast ett bra sätt att införa det nya systemet. Systemet prövas först på några entusiastiska användare. När de accepterar systemet, införs det i andra delar av verksamheten.

Godtagandetest görs för att säkerställa att systemet uppfyller alla krav i externspecifikationen och att den accepteras av användarna. En granskning efter implementering är sista steget av implementeringsfasen. Denna utförs för att avgöra om projektet möter alla mål i form av nytta och kostnader och för att göra rekommendationer inför framtiden.

Drift och underhåll

Fasen drift och underhåll börjar efter att användarna har accepterat det nya systemet. Den består av två aktiviteter: *drift och support* samt *underhåll*. Drift och support säkerställer att de tekniska delarna av systemet fortsätter att fungera

korrekt och att användarna använder det rätt och effektivt. Hur ofta man genererar summeringsrapporter och säkerhetskopierar databasen specificeras i en driftmanual. Ju mer användarna använder systemet desto mer brister upptäcks varefter förbättringar föreslås. Arbetet med att rätta dessa brister och förbättra informationssystemet kallas för *systemunderhåll*. (Alter, 1999)

3.3.2 Prototyp

En prototyp är en körbar modell av informationssystem som man skapar för att kunna testa sig fram när man inte vet exakt vad problemet och användarnas behov är. Man antar att användarna inte kan sätta fingret på vad informationssystemet borde klara av eller att de inte kan utvärdera en skriftlig specifikation. Med hjälp av en prototyp kommer man fram till vilka delar av systemet som måste utvecklas vidare och vilka som är överflödiga. Designarbetet av en prototyp är en iterativ process där man visar prototypen för användarna, får feedback och utvecklar efter den feedbacken, visar den nya versionen för användarna och så vidare.

Det finns två prototypvarianter: *throwaway* och *evolutionary prototype*. En throwaway prototype slängs när dennes funktion är uppfylld. Den är speciellt användbar när man vill jämföra flera olika alternativ, eftersom, att bygga prototyper är mycket billigare än att bygga hela system. En evolutionary prototype byggs för att vidareutvecklas och koder som man har skapat används i det nya systemet. Här bör man använda samma verktyg och program som man har planerat att använda i slutprodukten. (Alter, 1999)

Inledning

Inledningsfasen börjar med att användare och utvecklare fattar ett beslut att utveckla en prototyp, på grund av att de behöver mer kunskap innan de utvecklar slutsystemet. Användarna måste vara villiga och kapabla att ingå en iterativ designprocess. (Alter, 1999)

Utveckling

I utvecklingsfasen krävs snabb feedback. Man utvecklar en initial prototyp som klarar några funktioner, men inte alla. Sedan kommer man in i en iterativ process, där prototypen modifieras i varje iteration. Förändringar är baserade på användarnas kommentarer om föregående version. Tekniskt skiljer sig denna fas från samma fas i traditionell livscykelmodell. Här brukar man använda 4GL programmeringsspråk och verktyg, som man snabbt kan skapa program i, men som kanske saknar kapacitet för att göra stora datatransaktioner. När man sedan fortsätter med utvecklingen kan man välja att skifta till traditionell livscykelmetod eller att fortsätta och utveckla vidare på den koden man har producerat hittills. Skiftar man till traditionell livscykelmetod, producerar man specifikationer baserade på kunskap man har samlat med hjälp av prototypen. Det här görs om det nya informationssystemet är kritisk för verksamhetens utveckling. Om det däremot är till för att skapa rapporter eller om problemen förändras snabbt, är det bättre att utveckla prototypen vidare. (Alter, 1999)

Implementering

Delar av implementeringen utförs parallellt med utvecklingen. Användarna testar prototypen och blir vana att arbeta med den, så utbildningen för dessa användare brukar ta kortare tid. För användare som inte har medverkat i prototypdesign, är utbildningen lagd på samma sätt som i traditionell livscykelmetod. Konvertering planeras och utförs på samma sätt som i traditionell livscykelmetod. (Alter, 1999)

Drift och underhåll

Drift och underhåll för system skapade med hjälp av prototyper är samma som för system skapade traditionellt. (Alter, 1999)

3.3.3 Standardsystem

Eftersom det finns många företag med liknande behov och problem som löses med liknande informationssystem har det utvecklats *standardsystem* som säljes till företagen som paket av olika moduler. Att köpa ett standardsystem förkortar tiden som behövs för att få ett fungerande informationssystem och minskar utvecklingstiden för företaget. Hursomhelst, att välja och köpa ett standardsystem, kräver stor omsorg och företagets personal måste fortfarande gå igenom alla fyra faser för att säkerställa att det rätta standardsystemet har valts och att det har konfigurerats rätt. (Alter, 1999)

Inledning

Inledningsfasen kan börja med att man upptäcker ett problem, ser en utvecklingsmöjlighet eller att man får ett erbjudande från en standardsystemleverantör om att förbättra sin verksamhet med hjälp av deras system. Det är i alla fall bra att göra en funktionsspecifikation på samma sätt som i traditionell livscykelmetod. Detta underlättar valet av leverantör och system. Utan funktionsspecifikationen kan leverantören ha för mycket makt över hur företaget hanterar problemen och förändringsprocessen. (Alter, 1999)

Utveckling

Istället för att skriva en externspecifikation kallar man in leverantörer som presenterar sina produkter och demonstrerar vad de gör. Demonstrationer kan utföras i form av prototyper där några användare testar med verkliga data vad systemet kan göra. För att kunna jämföra olika system kan man använda *Request for Proposal* (RFP), som omvandlar den funktionella specifikationen till en checklista av krav. Leverantörerna förklarar hur deras system möter kraven. En bra leverantör undviker att sälja produkter som inte matchar företagets krav, eftersom det kan förstöra dess rykte. Det tar mycket tid och resurser att implementera ett dåligt matchande standardsystem. (Alter, 1999)

Implementering

Implementeringen börjar med att man bestämmer exakt hur systemet ska konfigureras och användas. De flesta standardsystem innehåller många valmöjligheter för att kunna tillfredsställa de olika kundernas behov. Oftast sätter man upp en provversion av systemet och jämför det med kundens krav. Leverantörens systemdokumentation brukar utvidgas med utbildningsmanual

utvecklad för specifik systemkonfiguration. Konvertering sker på samma sätt som i den *traditionella livscykeln*, med skillnaden att här är det leverantörens anställda och inte företagets som är experter på det nya systemet. (Alter, 1999)

Drift och underhåll

Drift och underhåll för standardsystem är samma som för traditionellt skapade system. Största skillnaden är att leverantören har största kunskap om systemet och är ansvarig för systemförbättringar. Leverantörer utvecklar nya *releaser* av standardsystemet och dessa baseras ofta på användarnas feedback. En release är en uppgraderad version av mjukvaran som kunden måste installera. Produktförbättringar är ett känsligt tema i leverantör-kund relationen. Leverantören får ofta långa listor av önskade förändringar och ibland inkluderar dessa listor förändringar av systemets källkod som leverantören inte gärna vill ändra. Leverantören och kunden jobbar ofta tillsammans för att identifiera de viktigaste förändringskraven men leverantören måste ibland förklara att många av de önskade förändringar inte kan göras eftersom de kan innebära stora risker. (Alter, 1999)

3.4 Utmaningar i systemutveckling

Typiska problem inom systemutveckling enligt Andersen, et. al. (1994) är:

- *Otillräcklig analys av arbetsprocessen och användarnas synpunkter* – Man måste ta reda på: Vem gynnar systemet mest? Vilka är användarnas behov? Vem är systemägare?
- *Processen är inte välgenomtänkt* – Alla delar av verksamheten och systemet bör definieras och presenteras genom modeller. Systemutvecklingsprocessen måste ha förankring i verksamheten.
- *Dåligt genomförda eller oavslutade tester* – Systemet ska testas innan det tas i bruk, eftersom otestade system innehåller många fel.
- *Orealistiska planer resulterar i förseningar* – Därför krävs en systematisk estimering av arbetets omfång.
- *Bristande kommunikation skapar problem och förseningar* – Det är en idealisk föreställning att systemutvecklingsprojekt kan förlöpa helt efter standard och rutiner. I verkligheten håller det oftast inte. Ibland kan rutinerna fungera rakt av, men ofta måste man experimentera för att överblicka och hantera situationen.
- *Prototyper börjar ofta köras som om de vore färdiga program* – Användarna tror ofta att prototypen är ett färdigt program och de vänjer sig snabbt vid den.

3.5 Kritiska framgångsfaktorer

Användning av kritiska framgångsfaktorer är ett sätt att identifiera några få saker som måste finnas för att tillförsäkra framgång på en strategisk nivå i en

organisation (Avison & Fitzgerald, 2003). Det är saker som man måste tänka på och säkerställa att de finns med för att verksamheten, organisationen eller projektet ska lyckas. Även om man börjar på en strategisk nivå kan dessa faktorer delas upp och tillämpas på lägre nivåer med mer specifika aktiviteter (Alter, 1999; Avison & Fitzgerald, 2003). I systemutvecklingssammanhang är det faktorer som är viktiga att tänka på för att minska risker och främja framgång vid utvecklingen.

Vid identifiering av kritiska framgångsfaktorer ska man undvika att identifiera för många, eftersom alla inte alltid är kritiska för framgång. Ett begränsat antal av dessa faktorer, grundligt analyserade, är därför något att sträva efter, menar Avison & Fitzgerald (2003).

3.6 Standardsystem som ett informationssystem

Inom begreppet informationssystem finns en del olika typer av system. Standardsystem eller affärssystem, internationellt kallat *Enterprise Resource Planning* (ERP), är ett sådant specifikt informationssystem. Det är en datorbaserad mjukvarulösning som har vuxit fram baserat på informationssystemens möjlighet att sköta företags hela transaktions- och informationsbehov (Hedman & Kalling, 2002).

3.6.1 Nyttan med standardsystem

Det är många företag som implementerar och använder affärssystem för att kunna hantera den allt mer växande mängden information. Företagens syfte med att anskaffa ett affärssystem är att skapa struktur i organisationen, effektivisera och underlätta arbetet. Meningen med affärssystem, även kallat standardsystem, är att de kan användas av flera olika företag eftersom de är utvecklade efter en standard som ska passa de flesta organisationers verksamheter (Ekman, 2004). Dessa system kan, enligt leverantörer, användas av alla företag oavsett storleken (Hedman & Kalling, 2002).

Standardsystem är uppbyggda av flera applikationer som stödjer affärsprocesser och som kan, om de integreras och används rätt, ge företagen stora vinster, genom att samordna och snabba upp processerna. Applikationerna kan även användas separat, men då kan nyttan gå förlorad. Ett standardsystem inkluderar vanligtvis applikationsområden som: *accounting*, *logistics* och *human resource management*. (Hedman & Kalling, 2002)

Enligt Hedman och Kalling (2002), anskaffar företagen standardsystem av flera olika anledningar, bland annat för att:

- Integrera finansiell information för att få bättre inblick i företagets ekonomi.
- Lagra kundinformation i ett system tillgängligt för flera avdelningar för att lättare koordinera order, produktion och utskick av varor.

- Standardisera produktionsrutiner och snabba upp processer genom automatisering.
- Standardisera personalinformation för att hålla reda på de anställdas arbetstider och förbättra kommunikationen inom företaget.

3.6.2 Utmaningar med standardsystem

Implementering av standardsystem är oftast komplex och kräver alltid omfattande konfigurationer för att systemet ska passa in i en speciell verksamhet (Hedman & Kalling, 2002).

Det är inte alltid så att införandet av ett standardsystem i ett företag garanterar den vinsten man har hoppats på. Organisationen är oftast den som anpassas till standardsystemet och inte tvärtom och organisationsförändring är viktigare än systemet i sig. Detta betyder att roller, arbetsuppgifter och arbetsrutiner förändras. Om människor som ska arbeta med det nya systemet inte är villiga att förändra sina sätt att arbeta, är risken stor att de motarbetar det och kräver stora förändringar av systemet. Detta kan resultera i dyra och utdragna anpassningar som gör systemet ostabilt, oanvändbart, dyrt och svårt att underhålla. Att få användarna att acceptera systemet och förändra sina rutiner är en utmaning och en av faktorerna som gör skillnaden mellan att lyckas eller misslyckas. (Hedman & Kalling, 2002)

3.7 Vidareutveckling

Det är svårt att finna litteratur som täcker vidareutveckling av informationssystem och speciellt vidareutveckling av standardsystem. De flesta metoder som rör systemutveckling fokuserar på nyutveckling eller utveckling från grunden. Vidareutveckling nämns oftast som en del av ett systems förvaltningsfas. (Wiktorin, 2003)

En typ av vidareutveckling av informationssystem är systemintegration eller sammankoppling av befintliga system. Systemintegration behövs när man vill ha stöd för verksamhetsprocesser som sträcker sig över olika affärsområden eller när man vill ha gränssnitt som kräver samverkan mellan flera system som jobbar i bakgrunden. (Wiktorin, 2003)

En annan typ är när befintliga system i företagen underhålls och vidareutvecklas och dessa aktiviteter svarar för två tredjedelar av informationssystemens kostnader på ett företag. Gamla system byts ut ofta genom stegvisa förändringar och utvidgas med nya funktioner som kan fordra introduktion av ny teknik. (Wiktorin, 2003)

3.7.1 VFS-modellen – en utvecklingsmodell i standardsystemkontext

Brandt et al. (1998) lägger fram en modell som de kallar för VFS-modellen (Välja och Förvalta Standardsystem). VFS-modellen är lämplig att användas vid val, utveckling och förvaltning av standardsystem på företag och den har kundens situation och kundens perspektiv som utgångspunkt. Den försöker svara på frågan: Hur ska man som kund välja och förvalta standardsystem för att nå en bättre verksamhet?

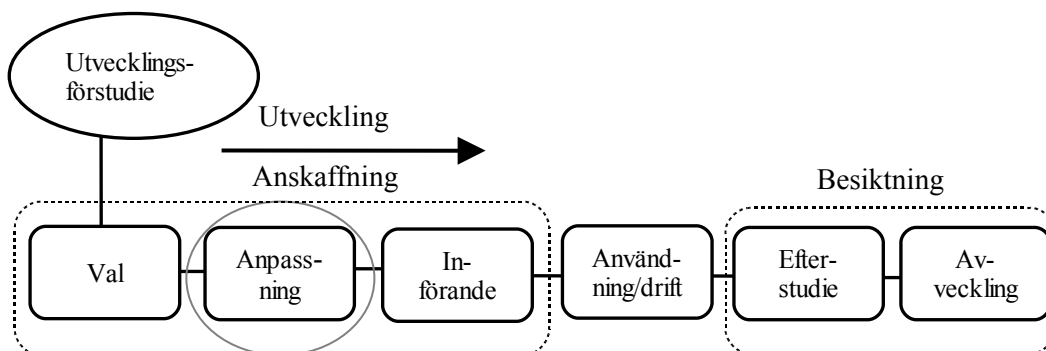
Vid anskaffning av standardsystem och utveckling av företagets verksamhet med hjälp av standardsystem har man speciella etapper som skiljer sig från vanlig systemutveckling. Dessa etapper berör val och anpassning. Vid vanlig systemutveckling genomför man en specificering och en utformning av ett helt nytt system från grunden (Brandt et al., 1998). En utvecklingsmodell i standardsystemkontext måste ta hänsyn till en jämförelse mellan verksamheten och flera standardsystem. Efter jämförelsen väljs det systemet som passar bäst och det anpassas ytterligare för att passa verksamheten exakt. En utvecklingsförstudie ligger till grund för arbetet med utvecklingen. Denna studie behöver peka på behovet av ett nytt standardsystem med syfte att förbättra verksamheten.

Enligt Brandt et al. (1998) består utveckling, med inriktning mot standardsystem, av ett antal faser. Faserna presenteras längre fram i teckning 1:

- *Val* – Valet ligger mellan flera, på marknaden tillgängliga, standardsystem som täcker verksamhetsområdet i fråga. Det systemet som lovar bäst stöd för verksamheten väljs.
- *Anpassning* – Efter att standardsystemet har valts, anpassas det till organisationen. Även organisationen måste anpassas till systemet.
- *Införande* – Implementering av det anpassade systemet görs i kundens verksamhet. Här gör man både installation och förankring i verksamheten.
- *Användning och drift* – Här menas ett kontinuerligt utnyttjande av systemet för att stödja verksamheten.
- *Efterstudie* – I en efterstudie värderas systemet med vikten på att fastställa huruvida systemet möter utlovade krav och ger effektivt stöd för verksamheten.
- *Avveckling* – Om systemet blir föråldrat eller olönsamt för verksamheten, avvecklas det. Detta leder ofta till en ny förstudie.

De tre första faserna kallas för standardsystemanskaffning. Med anskaffning menas upphandling eller återanvändning av system. De två sista faserna kallas för besiktning. Anskaffning, användning och besiktning är tre faser man kan urskilja i systemarbetet utifrån utvecklingsperspektiv. Faserna har vissa samband och normalt sett följs de i den ordning som nedan, men det finns speciella fall. Om man till exempel har ett visst standardsystem redan utpekat för användning i organisationen, hoppar man över valet och man går direkt från förstudie till anpassning. De faserna som alltid är med är anpassning och införande. Om man väljer standardsystem för flera olika områden samtidigt, kan det förekomma

parallella val med gemensamt eller åtskilt arbete för anpassning och införande. Inom koncernen kan det till exempel behövas ett standardsystem men med helt olika anpassningar till skilda bolag. (Brandt et al., 1998)

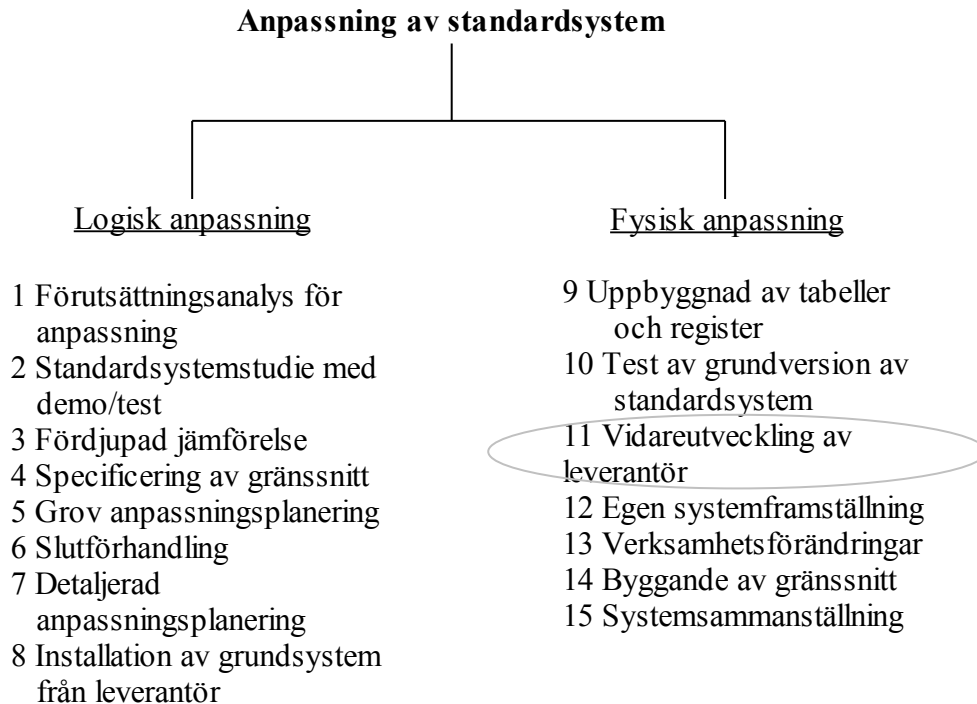


Teckning 1: Utvecklingen med inriktning mot standardsystem och dess faser (Brandt et al. 1998, s 109).

3.7.2 Anpassning

Under fasen *anpassning* sammanfogas verksamheten och det valda standardsystemet till varandra med målet att nå en tillräckligt god träffyta. Medan man under fasen *val* har flera standardsystem har man under *anpassning* endast det valda systemet kvar. Här görs fördjupande jämförelser mellan verksamheten och systemet. Hur mycket arbete som ska läggas ner på anpassningen beror på hur grundligt och ambitiöst man har genomfört valprocessen.

Fasen *anpassning* kan delas i två arbetsmoment: *logisk anpassning* och *fysisk anpassning*. Logisk anpassning avser planering av hur man ska utnyttja standardsystemet på bästa sätt i verksamheten. I den logiska anpassningen utnyttjas resultat från förstudie och val. Fysisk anpassning består av realisering och sammanställning av olika delar av systemet till en helhet enligt en logisk anpassningsplan. Anpassningen kan utföras både under utveckling och förvaltning (Brandt et al., 1998). Arbetsstegen för anpassning av standardsystem presenteras i teckning 2.



Teckning 2: Arbetsstegen för anpassning av standardsystem där punkt 11 är inringad eftersom det är det området vi är inne på (Brandt et al. 1998, s 120).

3.8 Verktyg

Nedan följer en kortfattad förklaring av de verktyg som är vanligt förekommande inom systemutveckling i standardsystemkontext. Med verktyg menar vi i det här fallet både utvecklingsmiljöer och programmeringsspråk. Stycket bör ses som en guide för att öka förståelsen för innebörden av de verktyg som senare omnämns i bland annat de empiriska resultaten.

ABAP4

SAP R/3 applikationer är skrivna i programmeringsspråket ABAP. ABAP applikationer kommunicerar med det grafiska användargränssnittet och databashanteringssystemet i den centrala relationsdatabasen. (Internet: SAP Help, 20060405)

BEA WebLogic Portal

BEA WebLogic Portal är ett verktyg för utveckling av portalgränssnitt. Detta verktyg möjliggör snabb uppbyggnad av portaler vilka i sin tur ger en integrerad vy av applikationer, information och affärsprocesser. (Internet: BEA, 20060405)

Rational Rose XDE Developer

Rational Rose XDE Developer är en samling av modellbaserade utvecklingsverktyg för applikationer i J2EE och .NET. Rational Rose XDE Developer kan integreras med utvecklingsverktyg så som Eclipse IDE, IBM WebSphere Studio Application Developer, Integration Edition IDE och Microsoft Visual Studio .NET. (Internet: IBM, 20060405)

SAP NetWeaver Developer Studio

SAP NetWeaver Developer Studio bygger på ramprogrammet Eclipse med öppen källkod. Som utgångspunkt för alla Java-baserade utvecklingsverktyg och som en integreringsbas för alla infrastrukturkomponenter stödjer SAP NetWeaver Developer Studio utvecklingen av applikationer i Web Dynpro, Internettjänster och affärsapplikationer i Java/J2EE. Det stödjer utvecklingen av Java-projekt för bland andra SAP-tekniker, standardtekniker, Internettjänster samt J2EE. (Internet: SAP Sweden, 20060405)

Web Dynpro

Web Dynpro kompletterar standardverktygen för utveckling av användargränssnitt, inklusive inmatningskontroller, inmatningsstöd, stöd för flera språk, felhantering och buffringsmekanismer. Både SAP NetWeaver Developer Studio och Web Dynpro ingår i SAP Web Application Server som är en öppen infrastruktur för skalbara Internetapplikationer och -tjänster. (Internet: SAP Sweden, 20060405)

WebSphere Portal

IBMs WebSphere Portal är ett verktyg för portalutveckling genom sammanfogande av färdiga moduler, så kallade portlets. (Internet: ATENSIS, 20060405)

3.9 Sammanfattning av teorin

Här sammanfattas teorin som vi finner vara viktig för den kommande analysen och diskussionen. Sammanfattningen är sammanställd efter de styckeindelningar vi har i själva texten. Tabellen nedan underlättar att komma ihåg de viktiga delarna till fortsättningen.

Systemutvecklingsfaser	<ul style="list-style-type: none">• Inledning• Utveckling• Implementering• Drift och underhåll
Systemutvecklingstyper	<ul style="list-style-type: none">• Traditionell systemlivscykel• Standardsystem• Prototyp
Utmaningar i systemutveckling	<ul style="list-style-type: none">• Otillräcklig analys av arbetsprocessen och användarnas synpunkter

Systemutvecklingsfaser	<ul style="list-style-type: none"> • Inledning • Utveckling • Implementering • Drift och underhåll • Processen är inte välgenomtänkt • Dåligt genomförda eller oavslutade tester • Orealistiska planer • Prototyper som körs som färdiga program
Standardsystem	<ul style="list-style-type: none"> • ERP, Enterprise Resource Planning • Affärssystem
Nyttan med standardsystem	<ul style="list-style-type: none"> • Integration • Lagring • Standardisering
Utmaningar med standardsystem	<ul style="list-style-type: none"> • Implementeringen är komplex och kräver omfattande konfigurationer • Införandet garanterar inte alltid vinsten man hoppas på • Användarna kan kräva stora förändringar • För många anpassningar kan göra systemet instabilt
Vidareutveckling	<ul style="list-style-type: none"> • Systemintegration • Kostsamt
VFS-modellen	<ul style="list-style-type: none"> • Ur kundens perspektiv • Jämförelse mellan verksamheten och standardsystem • Modellens faser: val, anpassning, införande, användning och drift, efterstudie och avveckling • Logisk anpassning • Fysisk anpassning
Verktyg	<ul style="list-style-type: none"> • Utvecklingsmiljöer • Programmeringsspråk

4 Presentation av empiriska resultat

De empiriska resultaten presenteras här nedan genom separata berättelser för varje företag som vi har intervjuat. Dessa berättelser beskriver företagets verksamhet övergripande och med inriktning på vidareutveckling.

4.1 Intentia

Intentia är ett mjukvarubolag, vilket innebär att de bygger mjukvaror för att sälja affärssystem. Företaget är delat i två delar, en utvecklingsdel, *Intentia Research Development* (IRD), och en konsultdel, *Intentia Consulting* (IC). IRD är Intentias fabrik som bygger produkten och tar fram nya versioner av systemet. Andreas jämför IRD med gamla Sovjet som gjorde allt med en femårsplan. IRDs primära uppgift är att leverera så bra produkter som möjligt till IC men de får inte lov att leverera någon annanstans. IC är som gamla USA, säger Andreas: ”Har du bara pengar så kan vi göra precis vad du vill.” IC får bara sälja och implementera produkter som kommer från IRD, som är Intentias egna produkter. *Movex* heter Intentias egna affärssystem. Intentia har ungefär sex eller sju branscher som de anser att de är bäst i världen inom och de fokuserar bara på de branscherna. De är duktiga på det som man brukar kalla *Supply Chain*, det vill säga företag som är i *Make*, *Move* och *Maintain*-branschen vilket innebär att de tillverkar någonting, flyttar någonting eller underhåller någonting.

Vidareutveckling

Vidareutvecklingen fungerar så att kunderna ritar upp vilket system de vill köpa och i den bästa av världar så har man redan ett system som ser ut på det viset. Andreas säger dock att om man har 90% träffsäkerhet så är man extremt duktig, med andra ord så inträffar det väldigt sällan. Även om Andreas menar att alla affärssystemslieferantörer i världen är måna om att det ska bli så få anpassningar som möjligt så förekommer det ändå.

Andreas beskriver tre typer av anpassningar:

1. Den första typen av anpassning, *Anpassningstyp 1*, är den där kunden vill ha ett system som avviker lite från systemet som Intentia kan tillhandahålla, men det ryms ändå inom systemets ramar. Den typen av anpassning fångas redan upp i säljfasen av projektet. De anpassningarna ritas upp utifrån hur de processmässigt ska fungera och det finns en helt annan långsiktighet i hur man utvecklar den typen av anpassning.
2. Den andra typen av anpassning, *Anpassningstyp 2*, är när kunden till exempel har köpt upp ett nytt lager och behöver en anpassning som knyter ihop det nya med resten av systemet. I det fallet vill man inte bryta systemets flöde utan man bygger istället en applikation utanför systemet, en webbaserad lösning som till exempel en portal. Man har då ett webbgränssnitt ovanför själva systemets flöde dit man lyfter ut data och bygger en applikation som inte stör systemet. Det svåra med detta är att

man har en enorm datamängd som snurrar på det viset. Man ändrar väldigt lite på systemet för man måste ta hänsyn till strukturen som redan finns.

3. Den tredje typen av anpassning, *Anpassningstyp 3*, är när kunden kräver en så pass speciell anpassning att det i sin tur kräver att man går in i källkoden och anpassar produkten i kärnan

Anpassningstyp 1 och 2 är de vanligaste, säger Andreas och de enda som får göra anpassningar på Intentias system är de själva. Det fungerar precis som en garanti vid ett bilköp, även på ett system. Alla anpassningar som görs måste först kontrolleras med utvecklarna. I vissa fall händer det att man undersöker om det är en anpassning som fler skulle kunna vara intresserade av och då går man ut till stora kunder i samma segment och frågar dem om de vill ha det. Vill de det så kommer det att komma med i en senare release. Då är det ingen anpassning längre utan då har kunderna köpt en säkerhet på att de inte behöver betala för någon anpassning när de uppgraderar systemet.

Intentias projektmetodik vid vidareutveckling heter *Implex* och reglerar bland annat hur man sköter dokumentation, projektstarten där man ritar upp processerna och flödet tillsammans med kunden och specificerar så mycket som möjligt, en designfas med utveckling där man de facto skapar koden och bygger systemet och testning och verifiering där produkterna alltid testas tillsammans med IBM. De olika faserna genomgås lite parallellt med varandra. Till exempel sker dokumentation under hela projektet.

I inledningen finns det tre upphandlingsformer, berättar Andreas:

1. Den första fungerar så att kundföretaget bjuder in stora leverantörer, så som Intenia, till en Offertförfrågan. Då har man med sig till exempel en *Senior Adviser*, en *Solution Architect* som är expert på processer, *Seniorkonsulter* och liknande. Sedan skriver de en lång dokumentation och har möten med cheferna på kundföretaget där man visar upp ett koncept som ger en bild av lösningen.
2. Den andra formen är när Intenia redan finns hos kunden och de redan kör systemet. Kunden gör en enklare förstudie, *Business Scan*, där de tittar på andra leverantörer samtidigt men Intenia har då ändå en förtur eftersom de redan är där. Då kontrollerar kunden med Intenia först om de kan göra det de vill ha för en kostnad som kunden har bestämt sig för från början.
3. Den tredje formen är när kunder lyfter in en extern part, en *uppdragstagare*, som köper systemet åt kunden.

Andreas berättar att man inte kan ändra på metoderna utan man ändrar istället på människorna, som inte alltid är så roligt, men det är så det fungerar. Det finns en uppdelning hos Intenia där konsulterna är indelade efter branscher och de är specialister inom det området. Bland utvecklarna finns inte den uppdelningen utan de går in och hjälper till oavsett var i processen de är. Med tiden blir det dock så att utvecklare som har jobbat länge blir duktigare på någon specifik applikation och då förekommer det även specialister bland utvecklarna. Arbetsteamet sätts ofta ihop med många olika specialister. Det ska gärna vara någon som kan processerna, någon som ritar upp arkitekturen så att den fungerar i en hårdvarumiljö. I större projekt brukar det vara med en arkitekt som håller ihop det

rent tekniskt och om det är ett projekt med 20–30 personer så är det alltid med en tekniskt projektledare som ska hålla ihop projektet. Man skulle ofta vilja ha med någon som jobbar med design men i 99% av fallen så täcker inte budgeten detta.

En av anledningarna till att kunder kommer till Intenia är för att de har förtroende för dem. Intenia försöker därmed att bråka med kunden när de tycker att det kunden vill göra är fel eftersom man vill få ner antalet anpassningar. Anledningen till att man vill få ner antalet anpassningar hos Intenia, säger Andreas, är för att efter några år kan det bli problem att lyfta en massa anpassningar som kan kosta kunden lika mycket som systemet i sig. Genom att fokusera på branscher så som Intenia gör kan de satsa på att göra ett riktigt bra standardsystem inom det området. På det viset kan de även undvika anpassningar eftersom produkten blir bredare och täcker fler behov hos kunden. Man försöker även att tvinga kunden att hålla sig inom det skalet. Det gäller att hitta balansen genom att göra så mycket som möjligt av kundens budget samtidigt som man själv ska försöka överleva.

Intenia har även något som heter *Movex Express* som gör att man kan implementera i hög fart och det går då fortare att rita upp processerna och det går fortare att göra anpassningar. Det som är skillnaden är att produkten bantas ner från början och vill kunden ha några anpassningar så finns det en lista på ca 200 anpassningar, som Intenia har gjort tidigare, som de kan välja mellan.

Verktyg

Intenia utgår ifrån kundens behov när det gäller valet av verktyg, det vill säga, de använder det som är bäst utifrån kundens verksamhet. De har kompetens att ge kunderna precis vad de vill ha. Andreas uttrycker det så här: ”Vi använder vad man kallar *Best of the Breed*, alltså bäst på marknaden.” Till stor utsträckning använder de verktyg från Microsoft. *Movex*, Intenias affärssystem, är byggt i Java till 100%. Det gamla *Movex* var byggt i RPG och det finns 600 kunder kvar i Sverige som använder det ännu. IBMs AS/400 är väldigt vanligt hos kunder men även Microsofts NT Server och SQL förekommer.

Dokumentation, kvalitetssäkring och kritiska framgångsfaktorer

Dokumentationen är främst kunddriven. Kunderna vill ha full dokumentation på allt som händer i deras system. Dokumentationen är även ekonomistyrd eftersom det mesta hänger på hur mycket kunden är villig att betala för. Intenia följer ISO-standarderna för dokumentation och det innebär mycket bra dokumentationsmöjligheter, säger Andreas. Inom metoden *Implex* dokumenteras allting, även för egen del. Detta görs främst för att vid händelse av att alla i en projektgrupp blir sjuka så ska det vara möjligt att sätta in en ny projektgrupp dagen efter som bara ska kunna fortsätta på projektet.

Intenia har valt att lägga ut delar av utvecklingen till Indien eftersom det är billigare. Detta gör att de har råd att ha fem personer som sitter och gör samma sak. De testar och verifierar varandra vilket leder till högre kvalitet på produkten. För framgångsfaktorns räkning är det viktigt att de verkligen arbetar med de processerna som kunden vill ha för tjänar inte kunden pengar så spelar det ingen roll hur bra system Intenia levererar, säger Andreas. Det är viktigt att Intenia kan identifiera kundens behov och att de gör rätt anpassningar till kunden, inte fler

men rätt anpassningar. Intenia försöker helt enkelt att föreslå lösningar inom systemets ramar. Det leder också till en högre kvalitet. Misslyckas Intenia med några kunder så får de inte fler kunder inom samma segment. Det är ytterst viktigt att man är överens med kunden från början, men även om man ändrar på planen så är det viktigt att man förhandlar om det.

Användarinvolvering

Det finns något som heter *Movex Användarförening* hos Intenia. Det är helt och hållet kundernas organisation där de är med utifrån sin bransch och har möten ofta för att sitta och diskutera vad de skulle vilja ha med i systemet, vad de tycker är bra eller dåligt och så vidare. De kraven skickas sedan direkt till IRD som till viss del jobbar utifrån de kraven. I övrigt involveras inte användarna i projekten eftersom pengarna styr och att involvera användare i test och liknande kostar mycket pengar och tid. Feedback på systemet kommer tillbaka mer som ett rykte. Till exempel om man har en bra användarmiljö eller inte eller om kundens krav inte uppfylls och så vidare.

Problem

Ibland händer det att det byggs en anpassning till systemet som inte bara slår igenom där det ska utan även i resten av systemet, vilket kan skapa stora bekymmer eftersom effekterna blir att det blir mycket svårt att hitta felet då det rör sig om en stor produkt med mycket kod. Det är oftast inget problem att bygga anpassningar, utan problemet kommer istället när man går tillbaka senare och ska lyfta alla tusentals anpassningar, vilket kan bli en mycket kostsam affär för kunden.

Det förekommer även att man inte kommer helt överens om hur systemet ska se ut när det blir klart, men även om man gör det så ändras bilden många gånger under processen. Till exempel att man har en överenskommelse om något från början men sedan när man kommer kunden närmare så upptäcker man saker som inte verkar stämma. Man undrar i det läget: ”Är det verkligen så ni vill ha det?”, säger Andreas. Till detta hör att köparna är inte användarna, vilket ibland kan orsaka problem. Köparna är en styrelse och de är främst intresserade av att spara pengar genom det nya systemet.

Det finns även vissa kommunikationsproblem ibland. Detta är framförallt när många specialister samlas i samma projekt. Dessa människor är ofta mycket olika och talar i princip olika språk även om de arbetar inom samma bolag. De har ju olika kompetensområden.

Ett annat problem som finns är att samma system ofta ser väldigt olika ut trots att det är samma system. Eftersom systemet byggs branschspecifikt så blir det automatiskt mycket annorlunda.

4.2 IBM

IBM är ett gigantiskt företag med 320 000 anställda. På SAP-avdelningen i Sverige jobbar runt 200 konsulter. För det mesta är IBM inblandade i ganska stora projekt och jobbar primärt mot stora multinationella företag. Men de är också närvarande på det som man kallar SMB – Small and Medium Businesses. Utifrån volymperspektiv är det 90% i de stora jättarna införande av affärssystem sker.

Vidareutveckling

Olle berättar att IBMs ERP-metod heter *Ascendant*, den bygger på SAP-metoden och används oavsett vilket standardsystem man arbetar med. *Ascendant* innehåller följande faser: *utvärdering, projektförberedelse, behovsspecifikation, realisering, produktionsförberedelse, produktionsstart* samt *drift och förvaltning*. Inom varje fas i ERP-metoden finns det dussintals aktiviteter och det är inte relevant för alla projekt. Man anpassar metoden även för att passa just ett visst utrullningsschema eller en bolagstyp. Ju mer unikt ett projekt är desto högre upp kliver man i metodstrukturen för att sedan skraddarsy det som är under strukturen. Ju mer standardmässigt det är desto längre ner i checklistorna kan man gå direkt och använda exakt det som man har gjort tusentals gånger innan.

Inledning: Förstudie kan vara en del av försäljningsarbetet eller någonting som kunden redan har gjort. IBM vill dock helst vara med och göra förstudien för de vet vad det är som behöver göras i en förstudiefas för att få det optimala resultatet i implementationen. I projektförberedelsefasen, gör man en processkartläggning där man gör nulägesanalys, affärsnyttoanalys och en processkarta (Business Blueprint) som är en ritning för hur systemet och verksamheten ska fungera. I behovsspecifikationsfasen tar man fram en detaljerad dokumentation över projektet där alla krav på processer och applikationer definieras. En eventuell demonstration av systemet sätts upp i denna fas.

Utveckling: Realisering är den fas då man utvecklar systemet. Utvecklingsarbete görs även delvis i kartläggningsfasen då konsulterna börjar prova och testa vad som går att göra.

Implementering: Här skapas en grund att stå på och sedan kör man in en pilot i en verksamhet. Produktionsförberedelse är den fas då testning sker parallellt med utbildning och förberedelser för produktionsstart. Produktionsstart består av datamigrering och är det sista steget innan man trycker på startknappen i det nya systemet.

Drift och förvaltning: Den sista fasen, med Hyper Care Support, det vill säga att de som har varit med och utvecklat är nu tillgängliga för support. Senare lämnar man över till en framtida supportorganisation som ska äga supporten.

Olle och Peter berättar att om det görs anpassningar så är det oftast IBM, eller deras konsulter som utför dem. IBM sköter utvecklingsdelen av projektet, det vill säga från det att man har bestämt sig att köpa ett standardsystem till det att man driftsätter det. IBM har en metod som beskriver hur de ska hantera förändringsbegäran under vägen. När det gäller metod är det i princip samma tankesätt de använder i alla projekt, oavsett om det är en portalapplikation eller ett standardsystem.

Utveckling av tilläggsapplikationer bestämmer man sig för redan i kartläggningsfasen. Om standardsystemet inte bär den funktionaliteten kunden vill ha eller om kunden inom ett visst område vill ha någon tilläggsfunktionalitet, sätter man sig och ritar som i vilket systemutvecklingssammanhang som helst. Här bestämmer man till exempel hur gränssnittet ska se ut, säger Peter. Det är möjligt att man gör det genom att ändra källkoden i standardsystemet, men det är något som man helst inte vill göra. I vidareutveckling av standardsystem är det viktigt att man hamnar rätt från början, berättar Olle. Om det går snett, är det ofta därför att kunden inte vet vad den vill ha eller varför. Så det är viktigt att man definierar vad det är man ska uppnå och sedan hur man ska uppnå det. Har man gjort det ordentligt från början, blir själva genomförandet ganska smidigt. Men det som däremot ställer till problem och det som gör att konsultkostnader växer ibland, är att man vill ändra kraven senare under projektet. Man samlar därför processägare från olika geografier och olika bolagstyper och sedan sätter man sig ner med konsulterna till hjälp och hamrar fram processer och rutiner inom varje område från inköp till ekonomi, försäljning, lager, produktion och så vidare. Man söker svaren på frågorna: Hur ser det ut i dag? Vad det är man saknar? Om man har bestämt sig för ett system så tittar man på vad man kan göra inom ramen för det systemet. Detta kan ske tidigt i projektet, men man kan också inse detta under realiseringsfasen då man egentligen har processkartan framför sig och kommer på att man har missat något. Därför finns det i IBMs projekt en rutin för att hantera förändringsbegäran.

Innan man gör en vidareutveckling, analyserar man noggrant vad det innebär rent kostnadsmässigt för framtiden. Man vill säkerställa att det realiserar den nytta man är ute efter och hur den nytta står i relation till kostnaden. Med kostnader menas inte bara kortsiktiga kostnader för utveckling utan också merkostnader i den levande lösningen de kommande åren och vad det får för implikationer på lösningen i stort.

Tonvikten på hela projektet flyttas ofta ifrån IT till organisationsutvecklingen och man använder införandet av ett affärssystem till att göra en förändring i organisationen. Man vill ha samordning och liknande rutiner överallt och då använder man införandet av affärssystem som ett verktyg för detta.

Unikt för vidareutveckling i standardsystemkontext, berättar Olle och Peter, är att processen blir kortare och att det ska vara synkroniserat med det stora projektet. Man har precis samma livscykel fast det blir en mindre variant och sannolikheten är kanske större att man kan hitta och återanvända komponenter som redan finns någon annanstans i organisationen. De går genom processen i samma ordning just

för att de ska kunna checka av enligt IBMs checklista både ur ett införandeperspektiv och riskperspektiv.

Syftet med portalerna är att samla och integrera, säger Peter. Man samlar och integrerar applikationer och blandar information och applikationer från många olika källor. Logiken och interaktionen ska vara likartad mellan de här applikationerna. Man har gärna standardkonfigurationer i form av någon sorts rollhantering. Istället för att anpassa individuellt har man några specialkonfigurationer för användartyp A, användartyp B, användartyp C, som man kan ställa in och sedan möjligtvis i ytterligare ett steg anpassa till individen. Man ska göra det enkelt, för ju enklare och mer likartat man gör det desto lättare kan man införa nya saker och desto mindre behöver man lägga ner på träning. Användarna kan använda de grundläggande redskapen på ett enkelt sätt istället för att ägna sig åt de mer komplicerade och avancerade arbetsmomenten.

Verktyg

Man använder det verktyg som passar i sammanhanget och IBM har kompetens till att göra allt, säger Olle och Peter. Får de fritt spelrum, använder de naturligtvis helst sina egna verktyg om de löser uppgiften som krävs. Hur verktygen används mer specifikt skiljer sig från projekt till projekt, från företag till företag, från applikation till applikation eller från programspråk till programspråk. De styr inte det åt ett visst håll utan man väljer det som passar bäst just för det tillfället. Till exempel, SAP är uppbyggt i språket ABAP 4 och då är det ABAP 4 man använder sig av. WebSphere Portal är det portalverktyg som IBM helst vill använda. Men de har varit engagerade i projekt som använder WebLogic också. Från användbarhetssidan är man inriktad på att vara så klientoberoende som möjligt. Man försöker använda sådant som sker på servern för att användarna inte ska vara beroende av att ha konfigurerat sina datorer på rätt sätt, förklarar Peter.

Dokumentation, kritiska framgångsfaktorer och kvalitetssäkring

Inom IBM har man en projektledningsmetod som heter *Seven Keys of Success*. Här inkluderas alla de sakerna man ska ha kontroll på ur olika perspektiv allt ifrån omfattning till organisation. Seven Keys of Success används iterativt under hela projektlivscykeln och det är dem man utgår ifrån i veckostatusmöten, i månadsrapportering till projektledning och styrgrupp och så vidare, och det är likadant för hela IBM.

Riskhantering är en väldigt viktig del i den här typen av projekt. Det får inte gå fel, eftersom det är företagens kärna man pratar om. Därför är det väldigt viktigt att man redan från säljfasen bedömer, hanterar och motarbetar risker och att man har etablerade rutiner för det. I de fallen där projekten innebär väldigt mycket risk, om det är antingen en väldigt delikat balansgång eller väldigt stor offentlig exponering eller där det av någon anledning ser ut att gå fel, händer det att man från centralt håll skickar ut ett team som gör en projektrevision. Det är ett så kallat *Red Team* som har auktoritet att stänga projektet i fall det innebär en för stor risk för IBM eller för kunden.

Hur mycket och vad man ska dokumentera är en kostnadsfråga, säger Olle. Om man ska ha en oerhört stringent och detaljerad dokumentation, kostar det mycket

pengar, så det är också någonting som man sätter ambitionsnivå på i säljfasen. Det beror på vad kunden är villig att betala för i form av dokumentation och vem som ska göra dokumentationen. Man lägger ut så mycket man kan till kunden. Ur ren självbevarelsedrift är man dock väldigt noga med att dokumentera alla specifikationer och enskilda ändringar. Det nästan viktigaste av all dokumentation är vad det är man har kommit överens om att göra och hur man ska kunna bevisa att man har uppfyllt målet. Dokumentation förgrenar sig genom projektlivcykeln. Ett processkartläggningsdokument med nulägesbeskrivning är anpassat för att bli ett underlag för konfigurationsdokumentation. Det ska gå in i utbildningskanalen för att börja anpassas för att kunna bli utbildningsmaterial och det ska också anpassas så att det kan bli bra testdokumentation för de som ska testa systemet. Att slarva med avvikelседokumentation vid anpassning av ett standardsystem kan vara katastrofalt för framtiden. När man kommer till en uppgradering och inte kan göra en inventering på vilka anpassningar man har gjort innan, till exempel. När man trycker på startknappen gäller det att man rullar tillbaks de utvecklingarna som slår på den första releasen och ju längre fram man kommer desto mer rollbacks är det och desto mer utbildning av tidigare installationer behövs. Har man inte det så tappar man lätt kontroll för då finns risken att man har olika utvecklingsmiljöer i utrullningen och i supportkedjan.

Ett stort förändringsprojekt innehåller inte bara systemutvecklare, det innehåller branschkunigt folk, processkartläggnings- och processutvecklingsexperter samt *Change Management* experter. Change Management är i princip uppdelat i organisationsförändring, kommunikation och utbildning. Man tar in experter inom alla de här områdena. Att hitta experterna är inte svårt. Med hjälp av IBMs interna system, Blue Pages, i kombination med en stor global CV databas, kommer man i kontakt med alla 320 000 anställda och som Olle och Peter säger: ”Någonstans finns det någon som kan det.”

Användarinvolvering

Användare ska vara en integrerad del av projektet. En av anledningarna är att de är ambassadörer för förändringen och det är de som ska se till att systemet används på rätt sätt i framtiden. Man har även med processägare, som ska komma från verksamheten och det ska helst inte vara vem som helst. Får IBM välja så ska det vara experter inom vart och ett av processområdena. Dessa personer skall kunna bestämma om man ska göra anpassningen eller om standard-funktionaliteterna ska användas för att de är tillräckligt bra.

Problem

Portaler är inte lösningen på alla problem, som många användare tror. Det hänger ihop med hur engagerad man är i att använda portalen. Så fort man gör så att någon måste anpassa portalen för att kunna använda den, har man byggt upp en tröskel som i sin tur kräver ett större engagemang och större intresse av användarna. Om man inte har det större intresset slutar man att använda det. Därför bör man ha en sorts standard som man sedan har möjlighet att anpassa endast om man vill.

Det klassiska problemet är att kunden vill ha något som ställer till det eller som innebär ett merarbete både i utvecklingsfasen och driftsfasen och som man anser

har ett för högt pris för verksamheten i relation till nyttan. Man ska ha en avvägning mellan standard- och speciallösning.

Ett problem kan vara att en leverantör säger ja till alla anpassningar men man fokuserar inte särskilt mycket på affärsnyttan. Olle och Peter säger att om man ”skruvar sönder systemet fullständigt”, kommer det att straffa kundföretaget i alla uppgraderingar och i alla framtida förändringar. Det man skapar är i princip inflexibla system. Hela nyttan med standardsystem är att de ska vara flexibla, det ska vara enkelt att uppgradera och infoga nya verksamheter eller kapa av verksamheter. Och ju mer man ändrar i systemet desto svårare blir sådana här aktiviteter. Det gäller att hitta balansen.

Beroende på vem som är projektägare kan det uppstå implementationsproblem. Om det är en lokal inköpsorganisation som äger projektet, vill de oftast ha det exakt så som de har tänkt sig och som passar deras verksamhet. De kanske sitter i ett markandsbolag i Brasilien, men deras sätt att hantera saker och ting kanske inte alls passar marknadsbolag i Tyskland. Och risken är att de då skruvar till det så att man inte kan realisera den lösningen i Tyskland. Det gäller att hitta någon sorts gyllene medelväg och rätt ambitionsnivå. Om man gör en global implementation av en portal eller ett affärssystem och det ägs på relativt låg nivå är det inte alls säkert att man har förankringen i verksamheten och det får effekter på användandet.

Ju högre ägandet av projektet är, om det är portalprojekt, stora integrationsprojekt eller stora affärssystemprojekt desto mer pragmatisk är ambitionsnivån. Högt upp har man förmågan att säga att det är summan av den totala nyttan som är det viktiga. Man struntar inte i den enskilde medarbetaren och att vederbörande kanske får det sämre, men om man är riktigt professionell i ett sånt här sammanhang kommunicerar man att: ”Detta kommer inte att bli bättre för alla, det kommer bli sämre för vissa, men totalt sätt så blir det bättre och företaget tjänar pengar”, säger Olle.

4.3 q2d Solutions

Den grundläggande affärsidén för q2d är att öka produktiviteten genom rollbaserade portaler, med andra ord, göra det mer *användarvänligt* för folk i sitt dagliga jobb med affärssystem genom att lyfta ut och sy ihop information från olika affärssystem eller från till exempel en Accessdatabas. Man vill kunna integrera allting i ett gränssnitt. Rollbaserade portaler fungerar som en vanlig webbläsare. Portalen innehåller information som är relevant för en viss persons roll inom företaget. Portalprogrammen innehåller ofta avancerad dokumenthantering, arbetsflöde, versionshantering och sökmöjligheter. Portalen kan dessutom användas som ett intranät. Användaren får tillgång till information baserad på sin roll i företaget och denne behöver bara logga in en gång för att få tillgång till alla system och dokument som denne har rättigheter till. Portalen kan också användas som ett extranät för att dela information mellan företag. En leverantör kan till exempel få information direkt från ett system genom att logga

in i extranätet. Allt detta leder till att anställda själva kan ändra tillåten data om deras lönekonton, föräldraledighet etc. Leverantörer kan uppdatera relevant data och kunder kan lägga order direkt i portalen.

q2d, med 15 anställda, är störst sedan 2001 i Skandinavien med erfarenhet av SAP Enterprise Portal. De är specialiserade på SAP Enterprise Portal och SAPs Netweaver teknologier men de har också kompetens inom Microsoft SharePoint Portal/.Net och IBM WebSphere Lotus/Domino. (Internet: q2d, 20051115)

Vidareutveckling

Utvecklingsprocessens faser är svåra att dela upp eftersom det i verkligheten inte alltid fungerar enligt några specifika faser. q2d arbetar snarare produktorienterat i de flesta fall även om det inte riktigt är meningen. I vissa projekt, som till exempel vid logistik går det att lägga upp processen på faser medan i andra görs projektstyrningen på underordnad nivå och det är inte samma sak som att detaljbygga upp något. I huvudsak är processen produkt- och teknikinriktad. Utvecklingsprocessen går enligt *Logikmetoden*, som Anders kallar det. Logikmetoden innehåller moment som till exempel en *kravspecifikation* som beskriver kundens mål och inriktning. Detta behövs för att q2d ska kunna koncentrera sig på det som kunden förväntar sig. Vidare består Logikmetoden av en form av *förstudie* med budgetering, dataflödesdiagram och en beslutspunkt, där man bestämmer huruvida man går vidare eller inte. Textinstruktioner är ofta svåra att förstå eftersom man inte ser sammanhang tydligt, menar Anders. Det är bättre med till exempel dataflödesscheman som visar processer och hur de hänger samman. q2d strävar efter att utföra fältstudier hos kunderna men för närvarande är det något som är för tidskrävande. Nästa del kallas för *projektering*. Kunden vill gärna se prototypen och det förekommer en tendens att man snabbt går över till *produktionsfasen*. Det sker aldrig i praktiken att man slänger prototypen, så som man lär sig i teorin. Kunderna tycker att det är svårt när de har sett något på skärmen. De tycker att det är färdigt då.

I vissa projekt är q2d med från början och då försöker de arbeta med de faserna inom systemutveckling som finns i teorin, till exempel enligt Logikmetoden. Detta påverkas dock oftast av budgeten. Kunden vill veta direkt vad det kommer att kosta och är oftast inte beredda att betala för någon förstudie om det inte kommer att resultera i ett vidarearbete. Detta kräver att man har erfarenhet och kan titta i pappret och se en cirka tid för projektet. I andra projekt kommer q2d in i slutet på ett projekt där kunden säger hur det ska vara och tycker att q2d är experter och bör kunna skruva ihop det åt dem.

q2d har tillsammans med det danska företaget Invendium, som de samarbetar med, också utvecklat en metod som är portalspecifik. Det handlar om saker man inte ska glömma vid utveckling av en portal bland annat. De använder en struktur som heter PCD (Portal Content Directory) i vilken de administrerar uppbyggnaden av portalen enligt de arbetsuppgifter och den roll som användaren har. Här lägger man upp en lösning som man testar på skärmen och sedan bygger man vidare på den. Om man påbörjar en sådan lösning utan att ha tänkt igenom den ordentligt så är man illa ute. Det är därför viktigt att man har de portalspecifika hållpunkterna.

Verktyg

De verktyg som q2d använder vid utvecklingen av portaler är bland andra SAP verktygen, SAP NetWeaver Developer Studio och WebDynpro. Även andra utvecklingsprogram går att använda för vidareutveckling av affärssystem, till exempel Microsofts SharePoint Portal eller IBMs WebSphere. Microsoft använder .Net för sin verksamhet, vilket är lite begränsat menar Anders. IBM och SAP bygger sitt i Java och Java är också det främsta programmeringsspråket hos q2d. Även .Net används hos q2d men inte i samma utsträckning som Java. Det är stor skillnad mellan traditionell systemutveckling och vidareutvecklingsprocessen som q2d driver, menar Anders. Han menar att det som gör vidareutvecklingen unik är de verktyg som q2d arbetar med vid vidareutvecklingen, till exempel WebDynpro. De gör hela processen mycket effektivare och snabbare. Alternativet är att bygga genom Java standarder som Eclipse, J2EE och JDE, som inte är några SAP produkter, men de är mycket mer tidskrävande. .Net ligger någonstans mitt emellan i effektivitet men det har q2d inte provat ut ännu, säger Anders.

Dokumentation, kvalitetssäkring och kritiska framgångsfaktorer

Dokumentation är det framförallt kunderna som driver eftersom projekten oftast ligger hos dem. I övrigt sker dokumentationen på sådant som är viktigt. Utvecklarna dokumenterar och kommenterar på nödvändiga ställen och förklarar hur de har gjort samt vilka delar som eventuellt kan vara problematiska och så vidare. Sådant som en normalerfaren portalkonsult ändå ska kunna dokumenteras inte specifikt. Eftersom projekten oftast ligger hos kunderna är det främst de som håller i trådarna för projektens kvalitet.

Den viktigaste kritiska framgångsfaktorn för q2d är den spetskompetens de har i ett jämförelsevis litet företag, säger Anders. Att kunna ta till sig den nya tekniken och använda den är en överlevnadsfråga för q2d. Den dagen de inte kan göra det kommer andra företag att ta över marknaden. Med andra ord ligger q2ds framgång i att de kan det senaste, med andra ord. Han säger att framgångsfaktorn för q2d är delvis också produktanknuten. q2d har egentligen andra kompetenser också, så som Data Warehouse, men där har de inte nått fram lika kraftigt. Att utveckla portaler var dock ett vitt område som q2d var med om från början och lyckades slå igenom med.

Det finns många stora företag som stampar i området för vidareutveckling i form av portaler och de vill antingen åt q2d eller så försöker de utbilda eget folk så mycket som möjligt. Exempelvis har q2d en man på Volvo som utbildar interna resurser där. Att utbilda sina konkurrenter bygger till stor del på att q2d ändå tror att företaget och verksamheten kommer att växa.

Användarinvolvering

I vissa projekt krävs det att utvecklarna går ut till kunderna för att visa och förklara systemet samt upplysa dem om vilka delar som finns. Mätning av användarhanteringen sker framförallt hos kundföretagen själva. q2d får inte reda på vad användarna tycker direkt eftersom de finns ett steg längre bak i kedjan och det användarna tycker filtreras genom någon intern avdelning på kundföretaget. På kundföretagen finns ofta en service och support avdelning men även här

filtreras en hel del av kundernas åsikter eftersom de som jobbar på den avdelningen även har varit med och ställt kraven på systemet efter vad de anser att deras användare vill ha. Den feedback som kommer till q2d sker då via företagens egna avdelningar. Anders menar att q2d är en för liten firma för att kunna göra direkta fältstudier hos användarna.

Problem

Den största och mest tidskrävande problemet för q2d är det faktum att tekniken hela tiden förnyas och misstagen som begicks på tidigare teknik sker återigen på den nya tekniken. Tekniken påverkar även sådant som att om SAP har någon bugg inne i systemet så kommer det fram via portalen och användaren tror då ofta att det är portalen det är fel på eftersom det är den som syns. I själva verket behöver det inte vara portalen alls som är felet. Ett affärssystem är oerhört stort och komplicerat och består av många delar som behövs hållas reda på. Om man bryter ner systemet består den av en mängd olika produkter och en del av dem är gemensamma för hela affärssystemet och därmed är det en otrolig mängd delar som måste hållas reda på. Portalvärlden har inte hängit med det som redan var byggt i affärssystemvärlden och har därmed inte hunnit så långt i förståelsen. Detta håller dock på att förändras och struktureras upp. En annan skillnad är också att affärssystemen har sitt eget system och det kontrollerar allting, vilket portalen inte gör på samma sätt. Portalen avslöjar inte vilka datorer den körs på i världen, vilken prestanda datorn har, om det finns brandväggar, vilket nätverk den finns på och så vidare.

4.4 Unikum

Unikum är ett företag som levererar affärssystem till mindre och medelstora företag. Systemet som Unikum utvecklar heter Pyramid och det körs idag av ca 3000 företag. Unikum arbetar ganska brett och har därmed inte nischat sitt system mot bara ett område. Företag som är kunder hos Unikum är bland andra handelsbolag och fackhandeldetaljister. Kunderna kör Pyramid utifrån något olika konfigurationer. Pyramid i sig är ett mindre antal huvudmoduler av typ sådant som alla företag behöver, redovisning, bokföring, projekthantering, eller produktionsstyrning och liknande. Normalt sätt ger Unikum ut en ny version varje år där de uppgraderar lite större saker, därmed finns det en kontinuerlig vidareutveckling av systemet.

Vidareutveckling

Det som vidareutvecklas är framförallt sådant som man kallar *skämmigheter*, det vill säga småsaker som man skulle vilja göra lite bättre i systemet. Kommer det in bra förslag på utveckling så vill man helst säga till kunden att: "Försök klara er så länge", säger Per, för det kommer att finnas med i nästa version om det ligger i linje med utvecklingen. Om kunden inte kan klara sig så gör man anpassningen.

Det finns en kraftig plattform, så som systemet är byggt i grunden och att förändra hur det är byggt är ett enormt arbete. En strävan hos Unikum är att det inte ska behövas några anpassningar men om det ändå måste göras så ska de göras så nära

kunden som möjligt och på ett så enkelt sätt som möjligt. När en ny version kommer ska det fungera med den. Det system som finns i grunden är byggt så att det ska kunna gå att ställa in det på väldigt många sätt.

Det finns *två vidareutvecklingstyper* hos Unikum. Dels är det de små anpassningarna och dels de lite större. De små är sådana som kan variera mellan att ta en dag till att ta en vecka att göra. Dessa processer är då inga stora processer utan det bara görs mer eller mindre. De lite större anpassningarna kräver dock lite mer. Där för man, till att börja, med *muntliga diskussioner* så att man är överens om vad som ska göras. Därefter skriver man en *specifikation* kring vad som ska omfattas av anpassningen. För att Unikum ska kunna sätta igång och arbeta så måste kunden sedan godkänna specifikationen och vara klar med alla regler och så vidare. Därefter sker *utvecklingen, testningen* och sist levererar man och *implementerar* anpassningen. De har börjat med att vara noga med att vid stora projekt bygga upp projektbeskrivningar och projektplaner med mål och delmål. Annars har Unikum inte direkt några namngivna metoder vid utveckling.

En strategi som de har är att de har ett *produktråd* som består av fem-tio personer som möts lite efter behov. Denna grupp har kontakt med marknaden och med återförsäljare så de tar in intryck den vägen till företaget.

Arbetsteamet för vidareutveckling ser likadana ut som vid vanlig utveckling. Vissa människor skiljer sig kanske lite genom att de är duktigare på att arbeta med grundstrukturen.

Den stora skillnaden mellan systemutveckling från grunden och vidareutveckling, menar Per, är den att man i vidareutveckling måste ta hänsyn till körande installationer.

Verktyg

På Unikum har man utvecklat egna verktyg och programspråk. De heter Visual Kosmos som är en utvecklingsmiljö och språket heter Beta. De använder utöver detta både en del Java och .Net.

Unikum har även utvecklat en tillägsmodul, en *design studio*, som är en integrerad funktion i systemet. Detta kan kunden köpa till och då får de möjlighet att vidareutveckla lite själva. Detta kan göras på två sätt. Det ena är att vanliga användare kan köra programmet och göra små förändringar, till exempel utseendemässiga förändringar. Det andra sättet är att konsulter kan lägga in lite programkod och liknande.

Dokumentation, kvalitetssäkring och kritiska framgångsfaktorer

Anpassningar dokumenteras för sig, även om det sker relativt lite, men i grunden finns ett stort hjälpsystem för användardokumentationen. Alla synpunkter och supportfel dokumenteras noga och dessutom kommer det en versionsdokumentation för varje ny version som utkommer. Väldigt mycket av dokumentationen sker i programmen i sig och det finns ett strukturerat sätt att hantera det på.

Kvalitetssäkringen ligger mest i testprocessen, säger Per. Upptäcker man ett fel så registreras det. Felen tar man hand om genom att en programmerare gör rättningen och en annan går in och gör en kontrolläsning av koden så att de kan säkra att de inte gör något fel. De anordnar även regelbundna kundmöten, som de kallar *kvalitetsmöten*, inom det teamen som har ansvar för de olika områdena.

Den kritiska framgångsfaktorn ligger främst i att Unikum inte har behövt byta utvecklingsfilosofi så fort det har tillkommit nya utvecklingsverktyg eller programspråk och så vidare. Eftersom de har sin egen utvecklingsmiljö och sitt egna språk som de har utvecklat själva så kan de gör som de vill och behöver inte följa strömmen.

Användarinvolvering

Unikum har relativt lite utbildning direkt mot kunderna utan istället har de satsat på att ha ett bra support- och serviceavtal direkt med kunden. Supportavtalet innebär dels att kunden får support och service och dels att de får nya versioner. De har daglig kontakt med sina kunder och de har ett system som stödjer detta genom att alla supportärenden finns loggade sedan flera år tillbaka. Dels är det så kallad *felsupport* som loggas och dels kan kunderna ge önskemål om förbättringar och liknande vilka registreras direkt i det systemet. Slut användarna är med en del vid de större anpassningsprojekten men inte så mycket i de mindre. Istället fungerar det så att man noterar synpunkter i systemet som både direkt kontakter och som statistik.

Problem

Problematiken med att förändra grundsystemet blir att man bygger två helt olika system och då får man en övergångsperiod som är väldigt svår ur många olika synvinklar. Oftast är det inte värt det eftersom den plattform som finns stämmer väldigt mycket med hur en affärsverksamhet bedrivs och med de data som finns där.

Kunskapsöverföringen mellan generationer kan ibland vara problematiskt, säger Per. När man ska göra nya saker och plockar in nya människor får de gamla oftast sitta och underhålla det gamla systemet. I det läget är det svårt att överföra kunskap mellan dessa.

En annan sak är att det är svårt att få korrekta datastrukturer som samtidigt inte är för arbetsamma, tycker Per, det är ett problem. Det blir ofta en kompromiss mellan effektivitet och en korrekt struktur.

Ett annat problem är att det lätt uppstår kommunikationsproblem av typen att kunden har en bild och konsulten en, men de är inte samma. Kunden tycker att vissa saker är självklara och säger därför inget om det och konsulten frågar inte eftersom denne inte tänkte på det alls.

5 Analys

I detta kapitel kommer uppsatsens empiriska resultat att granskas och analyseras med hjälp av teorin. Därigenom försöker vi skapa ett underlag för senare diskussion om ett ramverk för vidareutveckling. I föregående kapitel presenterade vi de empiriska resultaten separat för varje företag men här kommer resultaten istället att visas i kategorier och vi tar upp alla företagens synpunkter tillsammans. För varje stycke nedan så presenterar vi först de empiriska resultatet kortfattat i tabeller och därefter följer analysen. Vi vill poängtera att vi, i detta kapitel, främst diskuterar de delar av empirin som teorin stödjer samt de delar där empirin motsäger teorin. Av den anledningen nämns inte alla punkter som finns i tabellerna. Vi tycker dock att tabellerna är viktiga eftersom de sammanfattar empirin på ett överskådligt sätt.

5.1 Vidareutveckling

Företag	Vidareutveckling
Intentia	<p>De vill att det ska bli så få anpassningar som möjligt eftersom de ändå ska lyftas när man byter release.</p> <p>Ett sätt att undvika anpassningar är att bredda produkten inom en bransch och sedan tvinga kunden att hålla sig inom skalet.</p> <p>Tre typer av anpassningar: 1) de som sker inom systemet och bestäms redan i säljfasen, 2) de som sker utanför systemets flöde som en separat applikation, till exempel en portal och 3) de som kräver att man anpassar produkten i kärnan.</p> <p><i>Anpassningstyp 1</i> ritas upp utifrån hur det processmässigt ska fungera och det finns en helt annan långsiktighet i hur man utvecklar den anpassningen.</p> <p>Man bråkar med kunden om man tycker att kunden gör fel.</p>
IBM	<p>Man går inte så ofta från A till Ö i processen.</p> <p>Undviker förändringar och specialanpassningar eftersom man då frångår konceptet med standardsystem.</p> <p>Att ändra källkoden i standardsystemet vill man helst inte göra.</p> <p>Samma tankesätt som gäller oavsett om det är en portal, applikation eller standardsystem.</p> <p>Anpassningar och införande av tilläggsapplikationer sker INTE i slutet på en projektlivscykel utan utveckling av tilläggsapplikationer är någonting man bestämmer sig för redan i kartläggningsfasen.</p> <p>Projektlivscykler ser lite olika ut beroende på projekten.</p> <p>Projektlivscykeln i vidareutveckling är ungefär samma som i vanlig systemutveckling.</p>
q2d Solutions	Produktorienterad utvecklingsprocess.
Unikum	<p>Handlar om olika små saker som man förbättrar.</p> <p>En strävan är att det inte ska behövas anpassningar.</p>

Tabell 4: Kortfattad beskrivning av processerna i vidareutvecklingen

Företag	Unikhet i vidareutveckling
Intentia	<p>Man måste ta hänsyn till strukturen på grundsystemet när man bygger en tilläggsapplikation.</p> <p>De enda som får göra anpassningar på Intentias system är de själva.</p> <p>De kollar upp med stora kunder i samma segment om de också är intresserade av vissa anpassningar och i så fall tas de med i nästa version.</p> <p>De har en mycket djup branschkunskap.</p> <p>Vidareutvecklingsprocessen är den samma som för vanlig systemutveckling, det ligger bara olika fokus på de olika delarna.</p>
IBM	<p>Processen är kortare vid vidareutveckling.</p> <p>Sannolikheten är stor för återanvändbarhet av komponenter.</p> <p>Vidareutvecklingen ska vara synkroniserat med det stora systemet.</p>
q2d Solutions	<p>Verktygen är mycket effektiva för vidareutveckling av portaler.</p> <p>Någon förstudie finns inte vid vidareutveckling.</p>
Unikum	<p>I grundutvecklingen behöver man inte ta hänsyn till några körande installationer så som man behöver vid vidareutveckling.</p>

Tabell 5: Unikhet i vidareutveckling. En presentation av de saker som skiljer sig när det kommer till vidareutveckling

Företag	Problem
Intentia	<p>Det svåra med <i>Anpassningstyp 2</i> är att man har en enorm mängd data som snurrar utanför systemet.</p> <p>Anpassningar som någon byggt själv kan slå fel i hela systemet och det är ibland mycket svårt att hitta felet.</p> <p>Problemmet med anpassningar kommer egentligen först efter några år när man ska lyfta allt, vilket kostar nästan lika mycket som systemet i sig.</p> <p>Systemet kan se helt olika ut i olika branscher trots att det är samma system.</p> <p>Många specialister i samma arbetsteam kan skapa kommunikationsproblem.</p> <p>Balansgång mellan att göra så mycket som möjligt för kundens budget samtidigt som man själv ska tjäna pengar.</p> <p>Köparna är inte användarna.</p> <p>Bilden av systemet kan ändras hela tiden.</p> <p>Överenskommelse från början med kunden men vid närmare kontakt inser man att det kanske inte så kunden ska ha det.</p>
IBM	<p>Avvägning mellan standard och specialanpassningar, att hitta en gyllene medelväg.</p> <p>Att man skruvar sönder system efter kundens önskemål och skapar inflexibla system.</p> <p>Ofta har verksamheten dålig kontakt med IT-delen.</p> <p>När uppdragsgivaren inte definierar vad de vill ha och uppnå ordentligt från början.</p> <p>Olika behov och ambitionsnivå mellan olika aktörer.</p> <p>När konsulten kör projektet från A till Ö helt och hållet själv och sedan förväntar sig att man ska trycka på startknappen och allt ska fungera.</p> <p>Anpassningar av portaler kräver mycket engagemang och intresse i praktiken av användaren, annars slutar de att använda den.</p>
q2d Solutions	<p>Prototypen används direkt av användarna.</p> <p>Någon förstudien finns inte vid portalutvecklingen.</p> <p>Samma misstag som görs trots tidigare erfarenheter.</p> <p>Fel kravspecifikation från början skapar problem vid ombyggnad av portalen.</p> <p>Vissa problem kan ligga hos SAP men portalen drabbas med anklagelser först.</p> <p>Portalvärlden hänger inte alltid med det som redan är uppbyggt i standardsystemvärlden.</p>
Unikum	<p>Kommunikationsproblem och missuppfattningar, kunden har en uppfattning och konsulten en annan och man missar att ta upp sånt som är självklart för båda men ändå skiljer sig i grunden.</p> <p>Kunskapsöverföring från en generation till en annan.</p>

Tabell 6: En presentation av de problem som finns kring vidareutveckling

Vi har funnit att leverantörerna inte alltid kan genomföra alla vidareutvecklingsönskemål eftersom leverantörerna oftast är mer inställda på att hålla en standard i systemen och begränsa anpassningar, detta säger även Alter (1999). Hos till exempel Intentia väljer man att sköta sånt genom att ha ett slags vidareutvecklingsbibliotek. Med detta menar de att det finns en rad olika anpassningar som tidigare har utvecklats och som kunden kan få välja från istället

för att de ska utveckla ännu fler. Hos Unikum har man byggt systemet i grunden så att det ska kunna gå att ställa in på många sätt. På de här viset undviker man att behöva göra många anpassningar. Även på IBM diskuterar de mycket med sina kunder om huruvida de behöver alla anpassningar som de vill ha. Med andra ord så ser de här tre företagen på anpassningar som något som bör undvikas i så stor utsträckning som möjligt. På q2d däremot så ser man annorlunda på det. Deras verksamhet i sig är mer beroende av att utveckla anpassningar.

I fall användarna inte är villiga att förändra sina rutiner är risken stor att de motarbetar det nya systemet och kräver stora anpassningar som kan göra systemet ostabilt, oanvändbart, dyrt och svårt att underhålla (Hedman & Kalling, 2002). Olle på IBM säger att, om de skulle göra alla anpassningar efter kundens önskemål så skulle de kunna äventyra nyttan med standardsystem eftersom man istället kan komma att göra inflexibla system genom att: ”Skruva sönder för mycket”. Det som framförallt påverkar svårigheten med att göra anpassningar är att leverantörerna inte vill gå in och manipulera källkoden eftersom det är mycket riskfyllt (Alter, 1999) och detta bekräftar såväl Intenia som IBM, och även Unikum.

Enligt Wiktorin (2003) svarar underhållning och vidareutveckling av befintliga system i företagen för 2/3 av systemets kostnader. På IBM säger Olle och Peter att man väger nyttan i relation till kostnaden, och med kostnaden menas inte bara kortsiktiga utvecklingskostnader utan även merkostnader de kommande åren. Detta instämmer även Andreas på Intenia med när han säger att anpassningar kan bli en kostsam affär för kunden, även några år efter då många av dem måste lyftas. Dessutom garanterar inte införandet av anpassningen alltid den vinsten man har hoppats på (Hedman & Kalling, 2002) varför man på IBM till exempel poängterar att man måste ta hänsyn till nyttan av anpassningen.

En typ av vidareutveckling är systemintegration. Systemintegration behövs till exempel när man vill ha gränssnitt som sammankopplar olika system och lyfter fram information, enligt Wiktorin (2003). Andreas på Intenia berättar om tre typer av anpassningar varav en görs med hjälp av applikationer som utvecklas utanför själva grundsystemet, till exempel portaler. Även q2d och IBM använder detta sätt att vidareutveckla.

Brandt et al. (1998) säger att efter valet av standardsystem så behöver man göra anpassningar med målet att nå en tillräckligt god träffyta. Detta innebär att en viss mängd anpassningar måste ske. Andreas på Intenia berättar också att det sällan finns ett system som passar kundens önskemål helt perfekt, Han menar att om man har 90% träffsäkerhet så är man mycket duktig.

5.1.1 Metod

Företag	Metod
Intentia	<i>Implex</i> , som innehåller följande faser: <i>projektstart, utveckling, testning</i> och <i>dokumentation</i> .
IBM	Metoderna används mer som en checklista. Man anpassar metoden för att fungera på ett visst utrullningsschema eller en viss bolagstyp. Inom varje fas av metoden finns det tusentals aktiviteter och de är inte relevanta för alla projekt. <i>WWPMM, World Wide Project Management Method</i> ligger till grund för all projektmetodik i hela IBM. <i>GS Method, Global Services Method</i> har anpassats för att passa konsulter mer. <i>Ascendant</i> , ERP metod som bygger på SAP:s metod och innehåller följande faser: <i>utvärdering, projektförberedelse, behovsspecifikation, realisering, produktionsförberedelse, produktionsstart</i> samt <i>drift och förvaltning</i> .
q2d Solutions	Portalspecifik metod med hållpunkter för just portalutveckling <i>Logikmetoden</i> med följande faser: <i>förstudie, kravspecifikation, projektering</i> och <i>produktion</i> . Få konkreta vidareutvecklings specifika metoder.
Unikum	Inga namngivna vidareutvecklings specifika metoder.

Tabell 7: Övergripande metoder som används i vidareutveckling

Företag	Inledning
Intentia	<p>Vidareutvecklingsprocessen börjar redan i säljfasen. Utvecklarna måste alltid verifiera alla anpassningar innan de kan byggas.</p> <p>Kunden och konsulten ritar tillsammans upp processerna och flödet och kommer överens om hur de ska göra.</p> <p>Sätter upp så mycket specifikationer som möjligt vid start.</p> <p>Tre upphandlingsformer: 1) offertförfrågan, då flera leverantörer kommer till kunden och föreslår lösningar, 2) Intentia finns redan hos kunden och de kontrollerar med dem först innan de väljer en annan leverantör, 3) kunden tar in en uppdragstagare som köper systemet åt dem.</p>
IBM	<p><i>Utvärdering</i> sker i början och den kan vara olika lång beroende på hur mycket hemläxa kunden har gjort och den kan se helt annorlunda ut från fall till fall.</p> <p>Förstudie kan vara en del av försäljningsarbetet eller någonting som kunden har gjort redan.</p> <p>IBM vill helst vara med och göra förstudien för de vet vad det är som behövs göras i en förstudiefas för att få det optimala resultatet i implementationen.</p> <p>Samlar folk från hela världen – processägare – diskuterar fram processer och rutiner inom varje område.</p> <p>I <i>projektförberedelsefasen</i>, gör man en processkartläggning där man gör nulägesanalys, affärsnyttaanalys och en processkarta (<i>Business Blueprint</i>) som är en ritning för hur systemet och verksamheten ska fungera.</p> <p>I <i>behovsspecifikationsfasen</i>, man tar fram en detaljerad dokumentation över projektet där alla krav på processer och applikationer definieras. En eventuell demonstration sätts upp av systemet i denna fas.</p>
q2d Solutions	<p>Diskussioner med kunden leder till kravspecifikation.</p> <p>Kort förstudie med dataflödesdiagram, budgetering och beslutspunkt, text är svårt att förstå varför det bör vara i form av dataflöden eller annat som visar processer.</p> <p>Skriva direktiv om inriktning och mål med projektet.</p>
Unikum	<p>Muntliga diskussioner.</p> <p>Tar fram en specifikation av omfattningen av anpassningen. innehållandes projektbeskrivning, projektplan med mål och delmål.</p> <p>Kunden behöver godkänna specifikationen samt reglerna kring testningen.</p>

Tabell 8: Inledningsfasen av vidareutvecklingsprocessen närmare beskrivet

Företag	Utveckling
Intentia	Man har en designfas där själva byggandet, utvecklingen av systemet sker.
IBM	<i>Realisering</i> är den fas då man jobbar i systemet på riktigt (görs även lite i kartläggningsfasen för konsulterna börjar prova och testa lite vad som går att göra).
q2d Solutions	Projektering med prototyp.
Unikum	Anpassningen utförs och levereras.

Tabell 9: Utvecklingsfasen i vidareutvecklingsprocessen

Företag	Implementering
Intentia	Testningen utförs alltid med tillsammans med IBM.
IBM	Skapa en grund att stå på och sedan kör man in ett pilotprojekt i en verksamhet <i>Produktionsförberedelse</i> är den fas då testing sker parallellt med utbildning och förberedelser för produktionsstart. <i>Produktionsstart</i> består av datamigrering och är i princip sista steget innan man trycker på startknappen i det nya systemet. <i>Pilotimplementering</i> som innebär att man testat systemet på några användare inom ett segment innan man rullar ut det i hela organisationen.
q2d Solutions	Implementering av prototypen.
Unikum	Lite utbildning.

Tabell 10: Implementeringsfasen i korthet

Företag	Drift och underhåll
Intentia	Support.
IBM	<i>Drift och förvaltning</i> är den sista fasen med <i>Hyper Care Support</i> det vill säga att de som varit med och utvecklat är nu tillgängliga för support. Senare lämnar man över till en framtida supportorganisation som ska äga supporten.
q2d Solutions	Support.
Unikum	Support.

Tabell 11: Kortfattad beskrivning av fasen drift och underhåll

Företag	Kvalitetssäkring och kritiska framgångsfaktorer
Intentia	IRD arbetar en del utifrån kraven från <i>Movex Användarförening</i> . Många anpassningar läggs ut där det är billigt att utveckla. De gör de processerna som kunden verkligen vill ha. Duktiga på att identifiera vad kunden vill ha. Duktiga på kundens branscher. Gör färre men rätt anpassningar. Man kommer överens från början och förhandlar om så fort något ändras.
IBM	Etablerade rutiner för riskhantering hela projektet. Seven Keys of Success – metoder för riskhantering som används iterativt under hela projektlivscykeln. Riskbedömning redan i säljfasen av projektet. Red Team som har befogenheter att stänga projekt som innebär för mycket risk för IBM.
q2d Solutions	Kunddrivna projekt. Spetskompetens. Snabba med att hantera den senaste teknik.
Unikum	Testprocess. Supportservice. Kontrolläsning av källkoden vid rättning. Regelbundna kundmöten inom de teamen som har ansvar för för de olika områdena. <i>Produktråd</i> som möts efter behov och har lite mer förbindelse med marknaden. Tack vare egna verktyg kan de behålla sin egen utvecklingsfilosofi när det kommer nya varianter.

Tabell 12: En presentation av kvalitetssäkring och kritiska framgångsfaktorer kring vidareutveckling

Företag	Kompetens och arbetsteam
Intentia	Specialister uppdelat efter branscher. Arbeteamerna sätts ihop av många specialister: någon som kan processerna, någon som ritar upp arkitekturen, eventuellt en arkitekt som håller ihop det rent tekniskt och om det är ett projekt med 20 – 30 personer så är det alltid med en tekniskt projektledare som ska hålla ihop projektet. Utvecklingsgrupp som bara arbetar med utvecklingen. Utvecklare kan efter längre tid också bli specialister.
IBM	Arbeteamerna kan bestå av projektledare, applikationskonsult, processverksamhetskonsult, användbarhetskonsult, <i>Change Management</i> experter för kommunikation, senior, branchinriktad SAP konsult, processkartläggnings- och processutvecklingsexperter. <i>Change Management</i> är uppdelat i organisationsförändring, kommunikation och utbildning med experter på varje område. Det finns alltid någon som kan det som behövs.
q2d Solutions	Samarbetar med ett danskt företag när kompetensen tryter.
Unikum	Arbeteamerna skiljer sig inte mycket i vidareutveckling. Sätter in folk som är extra duktiga på att bygga grunstrukturen.

Tabell 13: En presentation av kompetens och hur arbeteamerna kan se ut vid vidareutveckling

Det är sällan man går rakt igenom alla faser i systemutvecklingen (Andersen et al. 1994) och Olle på IBM bekräftar detta genom att berätta att man sällan går från A till Ö när man utvecklar och vidareutvecklar system. Anders på q2d säger att de inte har några tydliga faser alls. Vi ser några skillnader i utvecklingsprocessen hos de fyra olika företag som vi har intervjuat. Hos q2d till exempel kan vi urskilja användning av *prototyping* som främsta utvecklingssätt, medan hos de andra tre företagen ser vi att man följer en mer traditionell systemlivscykel, vilket innebär att man har ett strukturerat arbetssätt, även om man inte går från A till Ö.

Det sätt som q2d jobbar på, med prototyper, kan jämföras med det sätt som Alter beskriver, dvs *Evolutionary prototyping* som man bygger vidare på. *Throwaway prototyping* använder de sällan eller aldrig. Ett problem med prototyper kan vara att användarna börjar köra prototypen direkt eftersom de tror att det är en färdig produkt (Alter, 1999) och detta problem identifierar Anders på q2d.

Alter (1999) beskriver traditionell systemlivscykel som ett förlopp med förutbestämda steg och detta tycker vi att man kan urskilja i de metoder som de beskriver på IBM, Intentia och Unikum. På q2d finns i teorin en liknande metod men den följs mycket sällan eftersom man här är mer produktorienterad.

Alter (1999) pratar om att man vid utveckling av standardsystem kallar in leverantörer som presenterar sina produkter och demonstrerar med hjälp av prototyper vad deras system kan göra. På Intentia har man också en sådan upphandlingsform i början och redan där i säljfasen fångar de upp anpassningsönsknningar som senare ska vidareutvecklas, förklarar Andreas. Även på IBM säger man att man bestämmer sig för utveckling av tilläggsapplikationer redan i kartläggningsfasen. Andreas berättar att det inte är med hjälp av prototyper

som man presenterar systemet utan en enklare demonstration där man visar konceptet.

I inledningsskedet definieras förändringsbehov och man ska skriva en överenskommelse om hur förändringarna ska utföras (Alter 1999). På de företag, som vi har intervjuat, så gör man detta steg. De kallar det lite olika men själva arbetet ser likadant ut.

Alter (1999) säger att en bra leverantör undviker att sälja produkter som inte matchar företagets krav eftersom det kan förstöra dess rykte. Andreas på Intenia har samma åsikt och säger att det är viktigt att kunna identifiera kundens behov och att man gör rätt anpassningar. Man försöker helt enkelt föreslå lösningar inom systemets ramar. Om man misslyckas med några kunder så får man inte fler kunder i samma segment. Olle och Peter på IBM, säger också att det är viktigt att man hamnar rätt från början och levererar anpassningar som gör nytta i kundens verksamhet.

Enligt Andersen et al. (1994), skapar dålig kommunikation förseningar och problem. Per på Unikum har samma uppfattning. Det är viktigt att kunden och utvecklaren har samma bild av det som ska utvecklas. Ibland händer det att kunden är medveten om något men talar inte om det eftersom detta är så självklart för denne. Konsulten i sin tur tänker inte på det och frågar inte heller om det varför det då kan missas. Det kan även förekomma kommunikationsproblem mellan konsulterna som arbetar inom samma projekt, till exempel när det rör sig om specialister inom olika områden, säger Andreas på Intenia eller då det ska ske en kunskapsöverföring mellan olika generationer, säger Per på Unikum.

Implementeringsfasen, enligt Alter (1999), inkluderar planering, utbildning, konvertering från det gamla till det nya systemet samt uppföljning. På IBM till exempel har man utbildning i samband med implementeringsfasen.

På IBM anser de att det bästa sättet att implementera ett nytt system är genom att trycka på stoppknappen i det gamla systemet samtidigt som man trycker på startknappen i det nya eller genom en så kallad pilotimplementering. Dessa två sätt att implementera på talar även Alter (1999) om. Alter (1999) påstår även att det är lämpligt att köra det nya och det gamla systemet parallellt. Olle och Peter på IBM delar inte den uppfattningen. Det är riskfyllt att köra parallellt, vilket de menar har visat sig historiskt. Dessutom menar de att det är synnerligen dåligt ur förändringsperspektiv för då får man svårt att släppa hörnflaggan och hänger kvar i det gamla systemet, som de uttrycker det.

Alla företagen är överrens om att man under utvecklingsfasen utför själva arbetet med att bygga det som ska vidareutvecklas. Även Alter (1999) säger att det är under denna fas som design och programmering sker.

Alter (1999) säger att drift och support säkerställer att systemet fungerar som det ska och att användarna använder det på rätt sätt. Alla företagen i vår undersökning betonar att support är en viktig del av drift och underhåll.

Som vi har tolkat Unikums sätt att se på vidareutveckling tycker vi att det påminner om det som kallas drift och underhåll i standardsystemkontext enligt Alter (1999). Till exempel genom att de följer upp med nya releaser efter feedback från användaren. Vidareutvecklingen särskiljer sig sålunda inte som en egen utvecklingsprocess ur den synvinkeln utan förblir som en del av den ursprungliga utvecklingsprocessen. Alternativt kan man säga att den fasen av standardsystemutveckling som kallas drift och underhåll skulle kunna vara just vidareutveckling så som vi tolkar det.

5.2 Verktyg

<i>Företag</i>	<i>Verktyg</i>
Intentia	<i>Best of the Breed</i> , det som är bäst på marknaden. Man använder det som är bäst utifrån kundens verksamhet, det finns kompetens för allt som kunden önskar. Microsoftprodukter bland annat MS SQL. Java. IBMs AS/400. RPG. Rational Rose XDE Developer for Visual Studio.
IBM	ABAP 4 (vid SAP) och Java (vid Movex). WebSphere Portal och WebLogic (vid portalutveckling). Har kompetens inom alla verktyg.
q2d Solutions	SAP NetWeaver Developer Studio och Web Dynpro (SAP produkter) Java och .Net. Utvecklingsmiljöerna är snabbare än programmerare och hjälper att få upp användargränssnitt snabbt. Ibland är de dock lite stela och man kan inte få det att se riktigt speciellt ut alltid.
Unikum	Visual Kosmos (Egen utvecklingsmiljö). Java, Beta (Eget språk), .Net. Design Studio (tilläggsmodul integrerad i systemet) för egen (kundernas) påbyggnad och förändring av systemet.

Tabell 14: Verktyg som företagen använder vid vidareutveckling

De verktyg som alla fyra företag pratar om i vidareutvecklingskontext är antingen verktyg för utveckling av portaler eller för standardsystemutveckling i helhet. Inga av verktygen är typiska för vidareutveckling men några av dem, till exempel portalverktygen är mycket lämpliga eftersom de är effektiva och kan integreras på ett enkelt sätt med andra verktyg och utvecklingsmiljöer. På till exempel q2d säger Anders att verktygen, *SAP NetWeaver Developer Studio* och *Web Dynpro* (Internet: SAP Sweden, 20060405) ofta är mycket effektivare än om programmerare skulle skriva all kod från grunden.

5.3 Dokumentation

<i>Företag</i>	<i>Dokumentation</i>
Intentia	Kunddriven. Ekonomistyrtd. Dokumentationsmöjligheter enligt ISO standard.
IBM	Avvikelsedokumentation. Konfigurationsdokumentation. Dokumentation i självbevarelsedrift. Dokumentation av enskilda ändringar och specifikationer. Kunddriven dokumentation på grund av kostnaden.
q2d Solutions	Kunddriven dokumentation. Egen dokumentation av sådant som är viktigt.
Unikum	Synpunkter dokumenteras för statistik och förbättring. Hjälpssystem i grunden för användardokumentation. Strukturerad dokumentation i programmen i sig. Versionsdokumentation på vilka förändringar som har skett med en viss version. Små kundanpassningar dokumenteras relativt lite.

Tabell 15: Beskrivning av hur dokumentationen ser ut vid vidareutveckling

Dokumentation sker mer eller mindre under hela utvecklingsförloppet (Alter, 1999). På IBM förgrenar sig dokumentationen genom projektlivscykeln och anpassas en del för att passa på till exempel utbildning, support och konfiguration. Generellt sett så utför alla fyra företag dokumentation i någon omfattning. Framförallt är dokumentationen kunddriven eftersom det hänger på hur mycket kunden är villig att betala för i dokumentationsväg. Givetvis sker även någon form av dokumentation för eget bruk hos alla fyra, eftersom de själva vill kunna hålla reda på vad som har gjorts.

5.4 Användarinvolvering

<i>Företag</i>	<i>Användarinvolvering</i>
Intentia	<i>Movex Användarförening</i> som är en kundorganisation med möten där kunderna tar fram nya krav på systemet. Användare involveras inte specifikt i utvecklingen. Om det är bättre totalt sätt så låter man det vara så även om inte alla får det bättre.
IBM	Spegelbild av arbetsteamet från kundens sida med beslutsfärliga personer som kan ta beslut angående anpassningar eller standardfunktionalitet. Användarna ska vara en integrerad del av projektet. Användarna kan vara med som utvecklare i projekten vara med. Användarna kan vara med som superanvändare så att de utbildas i lösningen. Det är inte nödvändigt att det blir bättre för alla men om det blir bättre för de flesta är det bra nog.
q2d Solutions	För litet företag för att kunna involvera användarna. Nästan ingen kontakt med användarna. Feedback från användarna filtreras genom någon intern avdelning och når inte alltid fram direkt.
Unikum	Noterar alla synpunkter som kommer in både för förbättring och för statistik. Synpunkter når fram även via supportavdelningen. Slutanvändarna involveras till viss grad.

Tabell 16: Beskrivning av användarinvolveringen vid vidareutveckling

En förstudie bör vara användarorienterad, det vill säga att den bör ta hänsyn till användarnas synpunkter, menar Alter (1999). Detta följs sällan i praktiken eftersom kunderna oftast har en begränsad budget och den täcker inte användarinvolvering till den grad som önskas. Olle och Peter på IBM däremot säger att användarna är ambassadörer på deras projekt och de ska se till att systemet används på rätt sätt. Även på Unikum tar man viss hänsyn till användarnas synpunkter genom att dokumentera *felsupport*, som Per kallar det. Otillräcklig analys av arbetsprocesser och användarnas synpunkter, enligt Andersen (1994) skapar problem. Man ska kartlägga processer noga för att kunna förändra dem på ett gynnande sätt för företaget, säger man på IBM. Användarna ska involveras hela vägen, tycker de, för att det förbättrar kvalitén på den färdiga produkten om man tar hänsyn till vad användarna tycker. På Unikum och q2d håller man med. Man medger dock att användarna på Unikum är oftast inte med i så stor utsträckning om det rör sig om små projekt. På q2d och på Intentia är de inte alls med. Andreas på Intentia säger att: ”Köparna är inte användarna”. Det innebär att de kommer inte att använda systemet och köparnas ändamål är främst att köpa anpassningar som ryms inom deras budget. Hur användarna upplever det kommer tillbaka ryktesvägen senare, till exempel om användarmiljön är bra eller dålig eller om deras önskemål inte blev uppfyllda med de nya förändringarna. Dessutom säger de på IBM och Andreas på Intentia att det är den totala nytta som

är väsentligt. Det är inte säkert att förändringarna blir bättre för alla men om det är det för de flesta så räcker det, menar de.

5.5 Sammanfattning av analys

Här presenterar vi en sammanfattande tabell över de punkter som vi har tagit upp och diskuterat i analysen.

<i>Kategorier</i>	<i>Höjdpunkter i analysen</i>
Vidareutveckling	<ul style="list-style-type: none"> • Anpassningar bör undvikas • Varning för inflexibla system • Väga nyttan mot kostnader • Nå en träffyta • Systemintegration
Metod	<ul style="list-style-type: none"> • Sällan från A till Ö • Metod som liknar traditionell systemlivscykel • Olika grad av strukturering i metoden • Upphandlingsfas i inledningen • Överenskommelse • Identifiera kundens behov • Strävan efter en god kommunikation • Planering, utbildning, konvertering i implementeringsfasen • Pilotimplementering • Anpassningen byggs under utvecklingsfasen • Support
Verktyg	<ul style="list-style-type: none"> • Effektiva utvecklingsverktyg
Dokumentation	<ul style="list-style-type: none"> • Kunddriven • Avvikelsedokumentation
Användarinvolvering	<ul style="list-style-type: none"> • Köparna är inte användarna • Totala nyttan är viktigare än den enskilda individen

Tabell 17: En översikt av analysen

6 Resultat – Ramverk för vidareutveckling i standardsystemkontext

I detta kapitel presenterar och diskuterar vi resultatet av undersökningen. Vi återknyter till vårt syfte med undersökningen, nämligen att utveckla ett ramverk för vidareutveckling i standardsystemkontext. Inledningsvis förklarar vi vad som är unikt med vidareutveckling samt vilka problem som kan uppstå vid vidareutveckling. Därefter presenterar vi ett ramverk som vi har identifierat utifrån våra tolkningar av de empiriska resultaten samt teorin. I slutet av kapitlet diskuterar vi på vilket sätt vårt ramverk är relevant i standardsystemkontext.

6.1 Typer av vidareutveckling

Det finns tre typer av anpassningar som man vidareutvecklar:

1. När anpassningarna fångas upp i säljfasen och utvecklas parallellt med grundsystemet.
2. När kunden redan kör systemet men behöver en anpassning på grund av förändringar i verksamheten. Då utvecklar man en applikation som läggs utanför grundsystemet för att inte bryta systemets flöde, till exempel en portal.
3. När man måste förändra i själva grundsystemet eftersom kundens verksamhet kräver det.

6.2 Unikt med vidareutveckling

Man använder verktyg som underlättar och snabbar upp vidareutvecklingsprocessen. Det kan till exempel vara verktyg som leverantörer av stora standardsystem själva ger ut. Även om verktygen inte är ändamålsspecifika är de unika på så vis att de används mycket vid vidareutveckling och företagen anser att de utgör en stor skillnad.

Man måste ta stort hänsyn till körande installationer som man inte behöver göra när man utvecklar stora system från grunden. Detta gör att det även blir svårare att göra designförändringar.

Man gör inte alltid en förstudie inför projektet på samma sätt som vid traditionell systemutveckling. Processen är ofta kortare i helhet och sannolikheten är stor att man kan återanvända komponenter som finns någon annanstans.

6.3 Problem som kan dyka upp vid vidareutveckling

Ett problem som kan dyka upp är att man lyssnar för mycket på kundens önskemål och till slut skruvar sönder hela systemet vilket kan resultera i mycket inflexibla system. Systemen blir svåra att uppgradera och infoga i nya verksamheter eller så kan det bli svårt att kapa bort verksamheten från det anpassade systemet.

Om man skapar tilläggsapplikationer som behöver individanpassas så bygger man upp en tröskel eftersom det kräver ett större engagemang och större intresse hos användaren. Saknas intresset så slutar man använda det.

Det är ofta svårt att komma fram till en gyllene medelväg när det gäller avvägning mellan standard och specialanpassningar. Vidare är det även ofta så att olika aktörer hos kunden har olika behov och ambitionsnivå, vilket leder till att inte alla anpassningar kan införas i hela verksamheten och det gör att den totala nyttan inte blir lika stor som när hela verksamheten kan utnyttja den nya anpassningen.

6.4 Ramverk

Vi vill betona på nytt att vårt syfte med denna studie inte var att generalisera, alltså syftar alla våra resultat, som visas i detta kapitel, enbart på de företagen som vi har intervjuat. Vi presenterar våra resultat med att identifiera ett ramverk som vi visualiserar i Figur 6.1 nedan, *VUS*. Sedan kommer vi att gå in lite närmare på ramverkets specifika delar och diskutera dem djupare.

6.4.1 VUS

Överst i ramverket visas en *tankebubbla* med vilken vi vill visualisera att man, när man vidareutvecklar, måste ta hänsyn till befintliga standardsystem som körs i verksamheten. Därefter visas de *förändringsbehov* som startar vidareutvecklingsprocessen. Vidare ser man de fyra faserna: *inledning*, *utveckling*, *implementering* samt *drift och underhåll*. Slutligen kommer det ett *resultat* ur processen. Vid sidan av de fyra faserna visas de faktorer som påverkar processen genom hela förloppet. Eftersom det rör sig om VidareUtveckling av Standardsystem, så har vi valt att kalla den *VUS*.

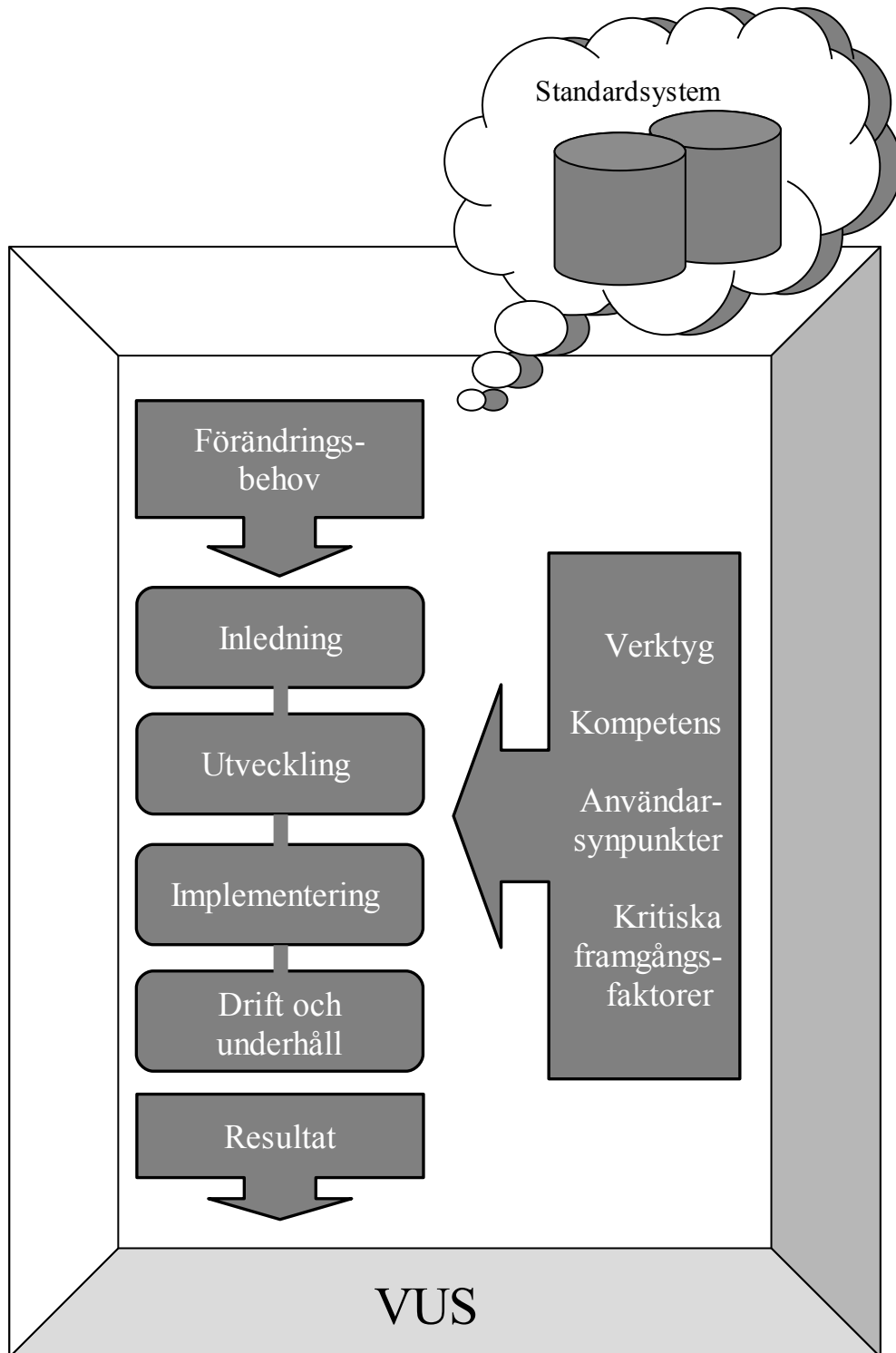


Illustration 1: VUS, VidareUtveckling av Standardsystem

6.5 Förändringsbehov

Man vidareutvecklar standardsystem för att den inbyggda funktionaliteten i dem inte alltid täcker kundens alla behov och önskemål med systemet. Man vill kunna nå en god träffyta mellan systemet och verksamheten. Ett annat syfte är att kundens verksamhet förändras och man vill följa förändringarna och stödja verksamheten i dess utveckling.

6.6 Metoden

Det är mycket liten skillnad på företagens metoder för vidareutveckling jämfört med systemutveckling från grunden. Företagen skiljer inte specifikt på vidareutveckling som en egen typ av utveckling utan ser det mer som en del av standardsystemutvecklingen, vilken i sin tur är en typ av systemutveckling. Systemlivsrytmen för vidareutveckling består av samma fyra faser som vanlig systemutveckling: *inledning, utveckling, implementering* samt *drift och underhåll*. Den största skillnaden ligger i hur tonvikten fördelas mellan faserna. Till exempel gör man sällan någon särskild förstudie i inledningsfasen vilket gör att den fasen blir kortare och man lägger istället mest fokus på att komma överens med kunden. Dessutom, i vidareutveckling lägger leverantören större fokus på att göra egen dokumentation eftersom det är viktigt att ha avvikelstdokumentation i ett senare skede. Den dokumentationen skiljer sig på så vis från den vanliga dokumentationen som finns med grundsystemet eftersom den beskriver specifikt de skillnader som finns mellan grundsystemet och det systemet som blir resultatet efter anpassningarna.

6.6.1 Inledningsfas

Inledningen av vidareutvecklingen sker i samband med säljfasen av grundsystemet. Redan där bestämmer kunden och leverantören vilka anpassningar som ska vidareutvecklas. Man skriver en specifikation och kostnader beräknas och vägs mot den totala affärsnyttan. I kostnadsberäkningen måste man även ta hänsyn till framtida kostnader i form av nya förändringar samt drift och underhåll. Här bör man börja ta hänsyn till körande installationer av grundsystemet. Det är dock väldigt viktigt att fokusera på det genom alla faser, även senare i livsrytmen. Vid diskussionen över vad som kommer att förändras i systemet behöver leverantören ibland säga ifrån när de tycker att kundens önskemål om anpassningar kan äventyra nyttan med standardsystemet i längden. Man ska försöka sträva efter att göra färre anpassningar men de som måste göras ska vara rätt. Leverantören vill inte utveckla lösningar som inte matchar kundens behov eftersom om det blir fel så riskeras leverantörens rykte vilket leder till en förlust av andra kunder.

6.6.2 Utveckling

Vidareutvecklingen kan ske parallellt med utvecklingen av grundsystemet men även vid senare tillfälle när kunden redan har ett körande system men vill lägga till delar på grund av förändringar i verksamheten. Leverantören anser dock att det oftast är mycket riskfyllt att gå in och manipulera källkoden i grundsystemet. Av den anledningen är det viktigt att försöka begränsa anpassningar som ska vidareutvecklas till det som är absolut nödvändigt för kundens verksamhet samt att försöka hålla dem utanför ramen för grundsystemet. Portaler är till exempel ett bra sätt att lyfta ut och samla information samt att integrera olika applikationer på. Genom dessa går det att presentera informationen på ett sätt som är användarvänligare.

6.6.3 Implementering

Implementeringen av de nya delarna görs med hänsyn till det körande systemet. Ibland kan det vara viktigt att göra en så kallad pilotimplementering där man testar anpassningen på ett urval av användare innan man rullar ut det i hela organisationen. Detta gör man för att förankra anpassningen i verksamheten. I samband med detta utbildas användarna till att kunna hantera de nya anpassningarna. Som underlag för utbildningen används dokumentation som görs under hela processen.

6.6.4 Drift och underhåll

Det är viktigt att leverantören har en supportverksamhet eftersom vidareutveckling kan resultera i ett annorlunda system. Eftersom det då inte längre är någon standardlösning så behöver kunden ha någonstans att vända sig vid problem. Genom att ha en sådan supportverksamhet kan leverantören även få feedback som hjälper dem att rätta eventuella fel och göra förbättringar.

6.7 Påverkande faktorer

Nedan beskrivs de faktorer som påverkar vidareutveckling av standardsystem vilka är: *verktyg, användarinvolvering, kompetens* och *kritiska framgångsfaktorer*.

6.7.1 Verktyg

De utvecklingsmiljöer som används vid vidareutveckling är framförallt: *BEA WebLogic Portal, Rational Rose XDE Developer, SAP NetWeaver Developer Studio, Web Dynpro* och *WebSphere Portal*. De programmeringsspråk som är vanligast är *ABAP 4, Java* och de som ryms inom *.Net*. I vissa fall kan det förekomma att man använder egenutvecklade verktyg som då bara används på det specifika företaget.

6.7.2 Användarinvolvering

Involveringen av användare varierar lite beroende på vidareutvecklingens omfattning. De är dock sällan direktinvolverade i utvecklingen. Deras synpunkter kan dock nå leverantören via supportverksamheten. Leverantören vill gärna ha slutanvändarna i projektet men det är en kostnadsfråga för kunden och det är kunden som bestämmer hur budgeten ska fördelas.

6.7.3 Kompetens

Arbeteameten särskiljer sig inte vid vidareutveckling utan det är ungefär samma människor som kopplas in på vidareutvecklingsprojekten som vid andra utvecklingsprojekt. Detta kan vara: *projektledare, teknisk projektledare, personer som kan processerna, applikationskonsulter, användbarhetskonsulter, samt någon från kundens sida alternativt någon med branschkunskaper.*

6.7.4 Kritiska framgångsfaktorer

De kritiska framgångsfaktorer som är avgörande för vidareutvecklingens kvalitet är:

- Att kunden och leverantören är helt överens från början om vilka delar som ska vidareutvecklas
- Att leverantören kan identifiera kundens behov
- Att man säkerställer att utvecklingen genererar nytta för kundens verksamhet
- Att kommunikationen fungerar bra mellan alla inblandade parter, såväl mellan kunden och leverantören som mellan konsulterna
- Att man planerar och utför testning och verifiering av de utvecklade delarna samt hur de samverkar med det körande systemet

6.8 Resultat av vidareutvecklingen

Resultaten som kommer av att man vidareutvecklar standardsystem är nyttan som kunden får ut av den. Då kundens verksamhet understöds med ett system som passar verksamheten kan kunden i längden tjäna på det och därmed finns det en nytta i vidareutveckling.

6.9 Ramverkets relevans i standardsystemkontext

Tidigare i teorikapitlet beskrev vi en standardsystemutvecklingsmodell, *VFS-modellen* (Brandt et al., 1998). Denna modell är presenterad utifrån kundens perspektiv. Enligt VFS-modellen har varje standardsystemutveckling en anpassningsfas och ett av stegen i denna fas är *vidareutveckling av leverantör* men

modellen beskriver inte närmare hur detta steg utförs. Det är där vårt ramverk, *VUS*, kommer in i bilden. *VUS* förklarar hur leverantörerna går tillväga vid vidareutveckling av standardsystem. Vårt ramverk kompletterar *VFS*-modellen genom att bidra med kunskap och ge en bredare bild av vidareutveckling ur leverantörens perspektiv. På så vis menar vi att vårt ramverk har relevans i standardsystemkontext.

7 Slutsats

Nedan för vi en slutdiskussion där vi återknyter till våra inledande frågeställningar och syftet med undersökningen. Vi diskuterar kvalitén och kunskapsbidraget i uppsatsen. Sist föreslår vi några alternativa synvinklar till framtida forskning.

I vår utgångspunkt till undersökningen hade vi väldigt få kunskaper om just vidareutveckling men vi förväntade oss att vidareutveckling skulle vara mer unikt än det har visat sig vara i förhållande till traditionell systemutveckling. Vi förväntade oss också att det skulle vara mer likt en egen systemutvecklingstyp. Efter undersökningen inser vi att det inte är så stor skillnad mellan metoder för vidareutveckling och standardsystemutveckling. Det föreligger endast några få skillnader vilka vi presenterade i våra resultat. Även våra intervjupersoner verkade ha svårt att se en klar och tydlig gräns mellan vidareutveckling och vanlig systemutveckling när de talade om sina utvecklingsmetoder.

Vårt syfte med undersökningen var att utveckla ett ramverk för vidareutveckling i standardsystemkontext. Detta presenterar vi i kapitel 6. Ramverket är ämnat att visualisera hur vidareutveckling ser ut idag hos några företag. Genom detta ramverk menar vi att man ska kunna förstå hur man kan gå tillväga vid denna typ av utveckling. Vi avser inte ramverket som en mall för vidareutveckling rakt av, eftersom det ännu endast är en början till en teori om vidareutveckling. Men vi menar att den skulle kunna ge en bild av praktiken som den är idag.

Svaret på vår första forskningsfråga ges genom presentationen av ramverket i det föregående kapitlet. Vi visar ramverket med en modell där de olika faserna i vidareutveckling presenteras: *inledning, utveckling, implementering* samt *drift och underhåll*. Dessutom visas de påverkande faktorerna, det vill säga de faktorer som mer eller mindre ingår i hela utvecklingsförloppet och påverkar resultatet. Vi visar även förändringsbehovet av vidareutveckling, samt resultatet, *nyttan*, som genereras av utvecklingen.

Svaret på den andra forskningsfrågan om vad som är unikt med vidareutveckling av standardsystem jämfört med traditionell systemutveckling är att man vid vidareutvecklingen använder effektiva utvecklingsverktyg som är lämpade för ändamålet, att man måste ta hänsyn till körande installationer och att utvecklingsprocessen har en kortare inledningsfas, vilket gör att även hela processen blir kortare.

Vi tror att vår undersökning bidrar med kunskap om vidareutveckling i standardsystemkontext eftersom vi visar hur det går till i praktiken när man vidareutvecklar standardsystem. Kunskapsbidraget i vår uppsats menar vi har relevans för systemvetenskapen eftersom vi fördjupar oss i ämnet informationssystem och bidrar med ny kunskap om ett hittills orört område. Dessutom hoppas vi att undersökningen kan väcka intresse för vidare forskning inom området.

Uppsatsens kvalitet har vi försökt säkra genom att dels koppla våra empiriska fynd till teorin och dels skicka materialet till företagen som vi har intervjuat så att de haft möjligheten att bekräfta om vi har använt informationen på rätt sätt. Dessutom har vi analyserat råmaterialet på var sitt håll så att vi inte skulle påverka varandras spontana intryck och tankar om råmaterialet. Sist men inte minst har vi beskrivit vårt tillvägagångssätt noga så att läsaren själv ska kunna avgöra om våra diskussioner är logiska eller inte.

7.1 Förslag till framtida forskning

Efter att ha genomfört den här undersökningen tycker vi att vägen har öppnats för en rad andra undersökningar som skulle kunna vara intressanta. Därför vill vi här ge förslag på infallsvinklar till andra studier.

Vår undersökning ger en bild av vidareutvecklingen från leverantörens perspektiv. Vårt förslag på en framtida forskning är att undersöka vidareutvecklingen från kundens perspektiv. Det skulle säkerligen fungera som ett bra komplement till den här undersökningen. I en sådan undersökning kan man fokusera på varför man väljer att göra vidareutveckling av ett standardsystem, vilka är påverkande faktorer och hur man utför dessa projekt.

På de företag som vi har intervjuat pratar man om hur kostnader för vidareutveckling vägs mot nyttan som man förväntar sig att de utvecklade anpassningarna ska generera på kundens företag. En intressant infallsvinkel skulle kunna vara att utföra en undersökning med syfte att ta reda på hur stora kostnader för vidareutveckling kan bli och om den nyttan och vinsten man hoppas på egentligen genereras.

Referenser

Litteratur

- Alter, S. (1999). *Information Systems: a management perspective*. Addison-Wesley
- Andersen, N.E., Kensing, F., Lassen, M., Lundin, J., Mathiassen, L., Munk-Madsen, A., Sörsgaard, P. (1994). *Professionel Systemudvikling: erfaringer, muligheder og handling*. København: Teknisk Forlag A/S
- Avison, D. & Fitzgerald, G. (2003). *Information Systems Development: methodologies, techniques and tools*. Berkshire: Mc Graw Hill Education.
- Brandt, P., Carlsson, R., Nilsson, A.G. (1998). *Välja och Förvalta Standardsystem*. Lund: Studentlitteratur
- Bryman, A. (1997). *Kvantitet och kvalitet i samhällsvetenskaplig forskning*. Lund: Studentlitteratur
- Denscombe, M. (2000). *Forskningshandboken: för småskaliga forskningsprojekt inom samhällsvetenskaperna*. Lund: Studentlitteratur
- Fitzgerald, B., Russo, N.L., Stolerman, E. (2002). *Information System Development: Methods in Action*. London: The McGraw-Hill Companies.
- Hedman, J., Kalling, T. (2002). *It and Business Models: concepts and theories*. Malmö: Daleke Grafiska
- Kaldrén, H. (1995). *Affärssystem*. Stockholm: Nordstedts Tryckeri.
- Kvale, S. (1997). *Den kvalitativa forskningsintervjun*. Lund: Studentlitteratur
- Miles, M.B., Huberman, A.M. (1994). *Qualitative Data Analysis*. London: Sage Publications
- Patel, R., Davidsson, B. (1994). *Forskningsmetodikens grunder; att planera, genomföra och rapportera en undersökning*. Lund: Studentlitteratur
- Svenning, C. (1999). *Metodboken*. Eslöv: Lorentz
- Svenning, C. (2003). *Metodboken*. Eslöv: Lorentz
- Wiktorin, L. (2003). *Systemutveckling på 2000-talet*. Lund: Studentlitteratur

Yin, R.K. (2003). *Case Study Research: design and methods*. Thousand Oaks, California: Sage Publications

Avhandlingar och uppsatser

Andersson, M., Hägg, D., Muratspahic, I. (2005). *Task Technology Fit/Användbarhet i affärssystem*. (Kandidatuppsats INF630). Lund: Lunds universitet, Institutionen för Informatik

Ekman, P. (2004). *Affärssystem & Affärsrelationer: en fallstudie av en leverantörs användning av affärssystem i interaktion med sina kunder*. (Mälardalen University Licentiate Thesis No.25). Västerås: Mälardalen University Press.

Hedman, J. (2003) *On enterprise systems artifacts: changes in information systems development and evaluation*. Lund: Lunds universitet, Institutionen för Informatik

Internet

ATENSIS

<http://www.atensis.se/www/site.nsf/page/produkter.html> (20060405)

BEA

<http://dev2dev.bea.com/wlportal/> (20060405)

Cern EDMS

<http://cedar.web.cern.ch/CEDAR/glossary.html#Framework> (20060408)

IBM

<http://www.ibm.se> (20060108)

<http://www-306.ibm.com/software/genservers/portal/> (20060405)

Intentia

<http://www.intentia.se> (20060315)

q2d Solutions

<http://www.q2d.se> (20051115)

SAP Help

http://help.sap.com/saphelp_46c/helpdata/en/fc/eb2c46358411d1829f0000e829bf/e/content.htm (20060405)

SAP Sweden

<http://www.sap.com/sweden/solutions/netweaver/webappserver/featuresfunctions/index.epx> (20060405)

Unikum

<http://www.unikum.se> (20051217)

Bilagor

Nedan följer ett antal bilagor. Intervjuguiden, informerat samtycke och ett utdrag ur transkriberingarna.

Intervjuguide 1

Detta är den första intervjuguiden som skrevs. Den gällde specifikt för q2d och frågorna var här helt anpassade till q2d. Efter intervjun tyckte vi att de behövdes ändras för att dels passa med fler företag och dels för att vissa frågor visade sig vara överflödiga. Denna blev senare ändrad vilket syns i bilaga 9.2.

Intervjuguide hos q2d – 20051122

Generell information

1. Namn
2. Sysselsättning
3. Position på företaget
4. Kontaktinformation

Intervju

1. Kan du beskriva q2d:s verksamhet och affärsidé?
2. Vilka typer av projekt jobbar ni med?
3. Hur ser processen ut från det att kunden beställer portaler till det att ni levererar och implementerar det?
4. Hur arbetar ni med vidareutvecklingen i standardsystemkontext?
5. Berätta om vilka faser processen består av?
6. Vilka verktyg och program använder ni?
7. Vilka utvecklingsmiljöer och programmeringsspråk använder ni er av?
8. Hur används verktygen mm specifikt?
9. Vilka strategier används för utveckling?
10. Vilka metoder och tekniker arbetar ni efter vid utvecklingen?
11. Om användarna involveras i projekten, hur går det till i så fall?
12. Vad anser ni är unikt med er utvecklingsprocess jämfört med att till exempel utveckla från scratch.
13. Hur dokumenterar ni utvecklingen? Finns den tillgänglig för oss?
14. Hur arbetar ni med kvalitetssäkringen? Vilka kriterier har ni för detta?
15. Vilka är de kritiska framgångsfaktorerna?
16. Har ni tidigare arbetat med vanlig systemutveckling, där man arbetar från scratch?
17. Är det ofta som ni missar att hålla deadlines? I så fall vad kan det bero på?
18. Vilka typer av problem generellt brukar ni stöta på i arbetet?

Intervjuguide 2 – Reviderad

Detta är den intervjuguide som vi använde till de tre övriga företagen, Intentia, IBM och Unikum.

Intervju

Generell information

1. Namn
2. Sysselsättning
3. Position på företaget
4. Kontaktinformation (telefon, e-post adress m.m.)

Intervju

1. Kan du beskriva företagets verksamhet och affärsidé kortfattat inom ramen för vidareutveckling/tilläggsutveckling? Vilka typer av tilläggsutvecklingsprojekt jobbar ni med?
2. Hur ser systemutvecklingsprocessen ut från beställning till leverans av systemet? Berätta om vilka faser processen består av?
3. Vilka strategier, metoder och tekniker arbetar ni efter vid utvecklingen?
4. Vilka verktyg, program, utvecklingsmiljöer och programmeringsspråk använder ni er av? Hur används dessa mer specifikt?
5. Hur ser arbetsteamerna för vidareutveckling ut? På vilket sätt skiljer de sig från andra team på företaget? Hur många arbetar inom teamen och vilka kompetenser har dem?
6. I hur stor utsträckning involveras användarna i projekten och hur går det till?
7. Vad anser ni är unikt med er utvecklingsprocess jämfört med vanlig systemutveckling? Kan du även beskriva med något exempel?
8. Hur sköts dokumentationen?
9. Hur arbetar ni med kvalitetssäkringen? Vilka kriterier har ni för detta?
10. Vilka är de kritiska framgångsfaktorerna?
11. Vilka typer av problem brukar ni stöta på i utvecklingsprocessen? Kan du ge exempel?
12. Hur brukar ni gå tillväga för att lösa de eventuella problem som dyker upp? Kan du ge något exempel?

Informerat samtycke

Här nedan finns avtalet om informerat samtycke som de intervjuade personerna på företagen skrivit på vid intervjutillfället.

Informerat samtycke

Vill du delta i en studie om affärssystem?

Syftet med studien är att ta fram ett ramverk för vidareutvecklingen i standardsystemkontext. Vi har för avsikt att genomföra studien genom att intervjua företag som innehar kunskap om ämnet.

Ditt deltagande i studien är helt frivilligt. Du kan när som helst avbryta ditt deltagande utan motivering.

Vi garanterar konfidentialitet genom att all information som presenteras sker med samtycke från företagets sida.

Ger du ditt samtycke av nedanstående datainsamlingsmetoder? Kryssa i det alternativ som du ger ditt samtycke till.

- Ljudinspelning
- Anteckningar

Genom att lämna din underskrift på detta avtal ger du ditt samtycke till ditt deltagande av studien.

Namn:

Ort:

Datum:

Ytterligare upplysningar om studien lämnas av nedanstående studenter:

Natalia Hajas
Kämnärsvägen 9 K:202, 226 46, Lund
046-211 28 73, 0733-50 69 13
natalia_hajas@hermes.ics.lu.se

Jasmina Ramicevic
Augustenborgsgatan 22A, 214 47, Malmö
040-96 50 97, 0736-58 18 23
jasmina_ramicevic@hermes.ics.lu.se