

SYSTEMLÖSNINGSMIGRATIONER TILL ÖPPEN KÄLLKOD

- En analys av utförda migrationsprojekt i offentliga förvaltningar sett utifrån ett riskperspektiv

Examensarbete inom huvudområdet datalogi
Grundnivå 15 högskolepoäng
Vårterminen 2012

Fredrik Ulvö

Handledare: Jonas Mellin
Examinator: Göran Falkman

Sammanfattning

Den här studien bidrar till att öka kunskapen inom området systemlösningsmigrationer genom att ta reda på vad som går att lära av tidigare genomförda migrationsprojekt. Studien genomförs genom att utföra fallstudier på avrapporterade migrationsprojekt som har utfört en migration från att använda proprietära systemlösningar till att använda systemlösningar baserade på öppen källkod. Efter en urvalsprocess kvarstod tre migrationsprojekt aktuella för analys utifrån ett riskperspektiv. Det var migrationsprojektet LiMux i Tyskland, ett projekt inom det franska gendarmeriet och ett projekt som utfördes bland skolorna i Bolzanoprovinsen i Italien.

Resultatet av studien visar bland annat att alla tre migrationsprojekten upplevde risken att inte kunna ta egna IT-beslut före den utförda migrationen. Detta berodde på att de före migrationen använde sig av proprietära systemlösningar, vilka de upplevt varit orsaken till detta. Den här risken säger de sig nu ha kunnat undvika genom att använda systemlösningar baserade på öppen källkod. De risker som förvaltningarna upplevt efter utförd migration kan sammanfattas med att det är risker som medför kortsiktiga problem för förvaltningarna. De räknar med att riskerna kommer att minska och problem kommer att lösas allt eftersom tiden går. Till exempel upplevde de att de har varit tvungna att anlita extern support för att lösa tekniska frågor relaterade till den nya systemlösningen. De säger dock att de i framtiden räknar med att själva ansvara för supporten när personalens kompetens har höjts inom förvaltningarna.

Studien visar också att en modell, BRW Migration Model for desktop OSS, kan användas för att minska riskerna vid en migrationsprocess. Detta görs genom att följa de nio steg som modellen beskriver som viktiga för en lyckad migrationsprocess.

Nyckelord: Offentlig förvaltning, öppen källkod, risker, migrationsprocess.

Innehållsförteckning

1	Introduktion.....	1
2	Bakgrund.....	3
2.1	Öppen källkod.....	3
2.1.1	Proprietär programvara kontra öppen källkod.....	3
2.1.2	Öppen standard och interoperabilitet.....	4
2.1.3	Systemlösning.....	4
2.1.4	Inläsningseffekt.....	4
2.2	Risk.....	5
2.2.1	Definition.....	5
2.2.2	Riskhantering.....	5
2.2.3	Ekonomi.....	6
2.3	Migrationsprocessmodell.....	6
3	Problemformulering.....	10
3.1	Syfte	10
3.2	Frågeställningar.....	10
3.3	Motivering	10
3.4	Avgränsningar.....	12
4	Metod.....	13
4.1	Genomförande.....	13
4.2	Fallstudie.....	14
4.3	Litteraturstudie.....	14
4.4	Validitet och reliabilitet.....	14
4.5	Alternativ metod.....	15
5	Resultat.....	16
5.1	Identifierade risker före systemlösningsmigration.....	16
5.2	Identifierade risker efter systemlösningsmigration.....	18
5.3	Analys efter migrationsprocessmodell.....	19
5.3.1	Münchens stad.....	19
5.3.2	Franska gendarmeriet.....	22
5.3.3	Bolzanoprovinsen.....	24
6	Diskussion.....	26
7	Slutsats.....	28
7.1	Framtida studier.....	28
7.2	Reflektion.....	29
	Referenser.....	30

Appendix

Appendix A: Urval – migrationsprojekt tillgängliga på Joinup

Appendix B: Münchens stad

Appendix C: Franska gendarmeriet

Appendix D: Bolzanoprovinsen

Figurförteckning

Figur 1: Mux.....	20
-------------------	----

Tabellförteckning

Tabell 1: Identifierade risker före systemlösningsmigration.....	16
------------------------------------------------------------------	----

Tabell 2: Identifierade risker efter systemlösningsmigration.....	18
-------------------------------------------------------------------	----

1 Introduktion

Öppen källkod innebär att den källkod som medföljer en programvara är fri för vem som helst att använda, förändra och distribuera vidare. Den öppna källkoden ska alltid medfölja tillsammans med det exekverbara programmet (Hansen m.fl. 2002). En av fördelarna med öppen källkod är att grundfilosofin är att källkoden ska vara kostnadsfri. Det finns dock inget som säger att den inte får säljas mot betalning (OSI, 1998a). En annan fördel med öppen källkod är att källkoden är tillgänglig för alla att modifiera efter sina egna behov. Detta gör det lättare att uppnå interoperabilitet, det vill säga att en förvaltnings olika IT-system lättare kan fungera tillsammans med varandra. Interoperabilitet är svårare att uppnå om källkoden är proprietär, det vill säga att den ägs av ett företag och därmed inte är tillgänglig för modifiering (Cerri m.fl. 2007). En risk med proprietära systemlösningar kan vara att information blir svårtillgänglig för medborgare eller andra avdelningar inom en förvaltning, om dessa inte nyttjar samma proprietära systemlösning. Ett exempel är kontorspaketet OpenOffice, som är baserat på öppen källkod. OpenOffice är gratis för alla att använda och den medföljande källkoden är öppen för modifiering efter egna behov. Detta gör det enkelt för en förvaltning att anpassa kontorspaketet till sin verksamhet samt för den enskilde medborgaren att ta del av förvaltningens information. Detta till skillnad från de proprietära kontorspaketen från Microsoft, som kostar pengar och som endast Microsoft själva kan modifiera.

Användandet av en proprietär systemlösning kan också leda till att det uppstår en inlåsningsseffekt inom förvaltningen. En inlåsningsseffekt kan uppstå när en förvaltnings systemlösningar till exempel inte stödjer öppna standarder och därmed skapar ett beroende av ett specifikt företag. Om förvaltningen i framtiden skulle vilja uppgradera eller tillföra extra funktionalitet till systemlösningen är sannolikheten stor att de måste välja det alternativ som det specifika företaget erbjuder (Simon, 2005).

Media rapporterar om hur bland annat offentliga förvaltningar runt om i Europa har kunnat göra stora ekonomiska besparingar genom att utföra systemlösningsmigrationer. Med systemlösningsmigration menas att till exempel en förvaltning byter från systemlösning A till systemlösning B. I fallen som media har rapporterat kring har förvaltningarna bytt från proprietära systemlösningar till systemlösningar baserade på öppen källkod. Gemensamt för förvaltningarna som har genomfört systemlösningsmigrationerna är att de bland annat säger sig kunna spara pengar och stärka förmågan att kunna ta sina egna IT-beslut med den nya systemlösningen baserad på öppen källkod (Joinup, 2012a).

På grund av det ökande intresset kring öppna källkodslösningar är det därför intressant att analysera dessa migrationsprojekt för att se vilka risker de har upplevt kopplat till systemlösningsmigrationerna. Detta med ambitionen att göra en sammanställning över vilka risker migrationsprojekten identifierat före och efter systemlösningsmigrationen samt hur de har gått tillväga i sina migrationsprocesser. Sammanställningen av analyserna ska kunna hjälpa beslutsfattare inom offentliga förvaltningar, framförallt IT-chefer, att få en överblick över de olika risker som är förknippade med den här typen av migrationsprojekt. Analyserna är begränsade till offentliga förvaltningar på grund av att det är svårare att få tag på analysmaterial från kommersiella företag som beskriver upplevda risker. Kommersiella företag med vinstintresse vill troligen inte delge potentiella problem i offentliga forum, då det skulle kunna leda till ett försämrat rykte.

Det här examensarbetet kommer alltså att analysera systemlösningsmigrationer i offentliga förvaltningar utifrån ett riskperspektiv. Fokus ligger på de förvaltningar som har migrerat

från en proprietär systemlösning till en systemlösning baserad på öppen källkod. Metoderna som används för att göra detta är dels litteraturstudier, med syftet att kunna definiera de viktiga begrepp som behandlas i studien och dels fallstudier på de avrapporterade migrationsprojekten. Syftet med fallstudierna är samla in information gällande upplevda risker inom migrationsprojekten.

De fallstudier som analyseras i den här studien är avrapporterade migrationsprojekt som finns tillgängliga på Europakommissionens samarbetsplattform Joinup (Joinup, 2012c). Joinup är en hemsida som är skapad för att publika organ lättare ska kunna utbyta information gällande olika typer av öppna källkodsprojekt. Efter att ha gjort ett urval bland tillgängliga projekt på hemsidan Joinup, fanns det tre projekt som matchade de framtagna urvalskriterierna. Ett migrationsprojekt i Tyskland, ett i Frankrike och ett i Italien där offentliga förvaltningar har utfört och avrapporterat stora delar av migrationsprojekten, vilket var vad ett av kriterierna krävde. Det tyska projektet går under namnet LiMux och pågår i Münchens stad. Alla administrativa arbetsdatorer inom stadens offentliga förvaltning ska migreras från att använda proprietära systemlösningar till systemlösningar baserade på öppen källkod. Det franska projektet är ett projekt inom det franska gendarmeriet, som är en del av det franska polisväsendet. De utför också en stor systemlösningsmigration. Det italienska projektet utfördes i Bolzanoprovinsen och omfattade systemlösningsmigrationer i 72 skolor. Förutom att de tre projekten matchade alla uppsatta kriterier har de även det gemensamt att de i sina avrapporteringar har uppgett att de är nöjda med resultaten av migrationen.

2 Bakgrund

Programvaror baserade på öppen källkod skapas av programmerare som är utspridda runt om i världen. Antingen på grund av egna initiativ eller för att kunna bidra med sin expertis i projekt med öppen källkod. Ett exempel på en öppen programvara utvecklad av frivilliga programmerare är webbserverapplikationen Apache. Apache introducerades år 1995 och är idag en utav de mest etablerade programvarorna för webbserverar (Lakka m.fl. 2012). Ytterligare exempel på populära programvaror med öppen källkod är operativsystemet Linux, webbläsaren Firefox och kontorspaketet OpenOffice. Gemensamt för alla dessa programvaror är att de är gratis för vem som helst att ladda ner och använda, utan kompromisser. Det spelar ingen roll om det är ett kommersiellt företag, offentlig förvaltning eller en privat hemanvändare som vill använda dem. Detta har gjort att intresset för att adaptera öppna källkodslösningar har ökat hos offentliga förvaltningar (Fuggetta, 2003).

2.1 Öppen källkod

2.1.1 Proprietär programvara kontra öppen källkod

Enligt Hansen m.fl. (2002) är proprietär programvara något som förknippas med en stängd källkod och som oftast endast är tillgänglig för dem som har utvecklat programvaran. När en programvara skapas görs det genom att programutvecklare skriver källkoden i ett programmeringsspråk, till exempel C, C++ eller Perl, som är lättare att förstå för människor än maskinkod. När källkoden är klar behandlas den av en kompilator. En kompilator är en programvara som gör om koden till maskinkod som är tillgänglig för datorns processorer. Det är den kompilerade koden som sedan levereras tillsammans med programvaran ut till kunden. När koden väl är kompilerad är den inte längre läsbar av människor. Detta innebär också att kunden inte heller kan modifiera koden även om den skulle vilja. När proprietära programvaror används brukar också användaren behöva acceptera olika avtal eller regler innan användning. Edwards (2005) visar ett exempel på ett sådant avtal; Microsofts EULA (Microsoft End User License Agreement), som är det avtal som medföljer en programvara som har inhandlats av Microsoft. I avtalet kan det till exempel stå att användaren inte har rätt att kopiera, distribuera eller modifiera programvaran.

Motsatsen till proprietär programvara är öppen källkod. Öppen källkod betyder inte bara fri tillgång till källkoden, utan också att programmet måste följa vissa kriterier (OSI, 1998b). Open Source Initiative är en icke vinstdrivande organisation grundad av Bruce Perens och Eric S. Raymond. Organisationen arbetar bland annat med att utbilda i samt förespråka användandet av öppen källkod. Det främsta syftet med organisationen är dock att underhålla något som kallas Open Source Definition. Open Source Definition är ett dokument innehållande en lista med tio olika kriterier som en programvara måste uppfylla om den ska kunna bli OSI-certifierad. Enligt Hansen m.fl. (2002) har det visat sig att de som inte väljer att certifiera sina programvaror med OSI-certifieringen kan få det svårare att konkurrera med dem som väljer att certifiera sig. Detta beror på att användare uppskattar programvara som följer de kriterier som det innebär att vara OSI-certifierad. Några av kriterierna som en programvara måste uppfylla är följande:

- Vara fritt för vem som helst att sälja eller distribuera programvaran vidare.
- Källkoden till programvaran ska medfölja eller på annat sätt finnas lättillgänglig.

- Programvaran ska tillåta användaren att modifiera koden samt distribuera vidare koden under samma villkor som originalet.
- Ingen diskriminering får ske, varken av personer eller användningsområde.

Dokumentet med de fullständiga kriterierna finns tillgänglig på organisationens hemsida för vidare läsning (OSI, 1998a).

2.1.2 Öppen standard och interoperabilitet

En öppen standard kan vara skapad av en ej vinstdrivande organisation som till exempel International Organization for Standardization (ISO). Organisationen har bland annat skapat den välkända standarden för ljud och video kompression, MPEG. Även större communitys, som The Internet Engineering Task Force (IETF), är delaktiga i att skapa och underhålla olika internetstandarder och protokoll. IETF är till exempel delaktiga i att underhålla kommunikationsprotokollen TCP/IP. En öppen standard kan också skapas av vinstdrivande företag som frivilligt väljer att göra standarden tillgänglig för användare och andra företag. Dock har de fortfarande rätten att i framtiden kontrollera den fortsatta utvecklingen av standarden. Exempel på en sådan öppen standard är Portable Document Format (PDF) som är utvecklad av Adobe Systems (Cerri m.fl. 2007).

Syftet med de olika formerna av en öppen standard är enligt Dalzien (2003) att skapa en interoperabilitet mellan olika programvaror och system. Dalzien säger också att en öppen standard gör att en användare inte nödvändigtvis behöver låsa sig till en specifik programvara eller system för att kunna använda en öppen standard. Så länge programvaran har stöd för den givna standarden är det inget problem att nyttja den vare sig det är en proprietär programvara eller en programvara baserad på öppen källkod.

Eftersom systemlösningar kan vara heterogena i förvaltningar blir interoperabilitet ofta ett krav, då systemlösningar består av olika operativsystem och applikationer. Om interoperabilitet inte uppnås i förvaltningen kan det medföra problem både för medborgare och avdelningar inom och utanför förvaltningen. Detta för att de har ett ansvar för att alla ska kunna ta del av informationen som produceras i de olika systemen (Statskontoret, 2003). Ett sätt att lättare uppnå interoperabilitet kan vara att använda sig av öppna standarder, för att slippa vara låst till ett specifikt filformat eller programvara (Cerri m.fl. 2007).

2.1.3 Systemlösning

Betydelsen av en systemlösning i den här studien kan variera från fall till fall. Detta för att kunna generalisera de olika programvaror som tas upp i migrationsprojekten där risker och migrationsprocesser analyseras. Systemlösning kan innebära hela den IT-plattform som en offentlig förvaltning använder för att drifva sina olika IT-system på. Begreppet kan då inkludera hårdvara, operativsystem, mellanprogram och applikationer. En systemlösning kan också vara en enskild programvara som till exempel OpenOffice. När begreppet systemlösningmigration används i den här studien är det alltid i form av en migration från en proprietär systemlösning till en systemlösning baserad på öppen källkod.

2.1.4 Inlåsnings effekt

När en förvaltning har en proprietär systemlösning som till exempel inte har stöd för öppna standarder kan det innebära att ett beroende skapas till en specifik leverantör. Detta betyder att om förvaltningen i framtiden vill uppgradera eller tillföra extra funktionalitet till systemlösningen ökar risken för att det enda alternativet som fungerar är systemlösningen

som den utvalda leverantören erbjuder. Detta kallas för inlåsningsseffekt (Simon, 2005). Om förvaltningen istället använder en systemlösning baserad på öppen källkod så ökar chansen till att modifiering och uppgradering lättare kan anpassas till systemlösningen. I systemlösningar baserade på öppen källkod påverkar varken källkod eller licenser valet av åtgärd.

2.2 Risk

För att kunna identifiera risker i ett migrationsprojekt måste först begreppen risk och riskhantering definieras. Detta görs enligt Pfleeger (2006).

2.2.1 Definition

Pfleeger (2006) säger att en risk är ett potentiellt problem som ett system eller användare av systemet kan drabbas av. För att kunna urskilja en risk ur ett projekts olika händelser finns det tre saker att titta på enligt Pfleeger, (2006):

1. *Att en förlust kan associeras med en händelse* – Händelsen måste generera en negativ effekt som kan vara till exempel: minskad säkerhet, förlorad tid, förlorade pengar eller förlorad förståelse.
2. *Hur sannolikt det är att händelsen inträffar* – Sannolikheten värderas genom att associera varje risk med ett värde. Det kan göras efter en skala där till exempel 0=omöjligt och 1=helt säkert. Om händelsen associeras med värdet 1 är sannolikheten för att händelsen kommer att inträffa helt säker. Det innebär att det finns ett problem i projektet som kommer att inträffa om inga åtgärder vidtas.
3. *I hur stor utsträckning det går att ändra resultatet av utgången* – Det måste gå att bestämma vad, om någonting, som kan göras för att eliminera eller minimera effekten av utgången.

2.2.2 Riskhantering

När väl en risk är identifierad finns det enligt Pfleeger, (2006) tre olika strategier för att hantera en identifierad risk.

1. *Undvika* risken genom att ändra de krav eller egenskaper som gäller för systemet.
2. *Överföra* risken till andra system eller tillgångar genom att till exempel köpa en försäkring.
3. *Acceptera* risken genom att kontrollera den med tillgängliga resurser och förbereda sig för att hantera eventuell förlust.

Den kostnad som associeras med resultatet av att en inträffad risk inkluderar även kostnaden för att minimera den. Detta leder fram till begreppet riskkostnadsgrad, som innebär skillnaden av riskexponeringskostnad delat med kostnaden för att minimera risken. Vilket kan illustreras med följande riskkostnadsformel:

$$\frac{(\text{riskexponering före reduktion}) - (\text{riskexponering efter reduktion})}{(\text{kostnad av riskreduktion})}$$

Om värdet av resultatet av formeln inte är högt nog bör andra alternativ eller reduceringstekniker ses över. För att ytterligare exemplifiera detta görs följande antagande: Om ett virus skulle drabba programvara X skulle riskexponeringen vara beräknad till 10.000

kr. Om ett virus skulle drabba programvara X efter riskreduktion i form av ett antivirusskydd beräknas riskexponeringen istället till 1000 kr. Om kostnaden av att implementera antivirusskyddet beräknas kosta 300 kr kan det se ut enligt följande riskkostnadsformel:

$$\text{(riskkostnadsformeln:)} \frac{(10.000) - (1000)}{300}$$

Detta skulle resultera i ett riskkostnadsvärde på 30. Vilket betyder att det skulle vara lönsamt att installera ett antivirusskydd eftersom det är långt över riskkostnadsvärdet 1, som är brytpunkten för när ett riskkostnadsresultat är plus minus noll. Det kan dock finnas scenarier där ett riskkostnadsvärde under 1 är acceptabelt. Till exempel kan en organisation välja att acceptera en ekonomisk förlust gentemot ett resultat av till exempel minskad inlåsnings effekt för en systemlösning. Det är alltså inte säkert att ekonomisk vinst alltid är motiveringen när en organisation väljer att utföra en systemlösning migration.

2.2.3 Ekonomi

Det finns ekonomiska risker med att göra en systemlösning migration. Förvaltningen riskerar till exempel att personalkostnaderna ökar på grund av att en kartläggning bör göras kring de resurser som kommer att behövas i samband med ett migrationsprojekt (Amant och Still, 2007). Existerande personal kan behöva avsätta tid som var planerad till annat eller så behöver konsulter hyras in för att se över behoven kring den nya systemlösningen. Andra utgifter kan vara ytterligare systemlösningar som behöver införskaffas för att uppnå interoperabilitet i systemet. Om förvaltningen behöver utreda vilka standarder som den nya systemlösningen kommer behöva ha stöd för kan interna eller externa personalresurser behöva införskaffas (Statskontoret, 2007). Om huvudanledningen till en systemlösning migration är att göra ekonomisk vinst, bör en kalkyl göras enligt den riskkostnadsformel som beskrivs av Pfleeger (2006). Detta för säkerställa att en tillräckligt hög riskkostnadsgrad går att uppnå med det tänkta systemlösning alternativet.

2.3 Migrationsprocessmodell

Om en förvaltning är intresserad av att göra en systemlösning migration är det viktigt att göra detta på ett organiserat sätt för att undvika problem. En modell för att kunna göra detta erbjuds av den migrationsmodell som Amant och Still (2007) skapat. Amant och Still (2007) har format migrationsmodellen efter att ha studerat tre olika migrationsprojekt i Sydafrika gällande öppna källkod migrationer i tre olika användningsområden. De tre olika områdena var utbildningssektorn, den kommersiella sektorn samt en statlig verksamhet. De hade alla valt att migrera hela eller delar av sina systemlösningar från proprietära systemlösningar till systemlösningar baserade på öppen källkod.

Den information som finns att tillgå gällande de tre fallstudier som Amant och Still (2007) analyserat är inte så detaljerad. I rapporten beskriver de framförallt det ena av de tre fallen. Det är den kommunala förvaltningen Mossel Bay, som är en relativt liten kommun i Sydafrika. Deras migrationsprojekt omfattade cirka 60 datorer som alla skulle migrera från att använda proprietära operativsystem och kontorspaket till motsvarande alternativ baserade på öppen källkod. Migrationen ansågs som lyckad och anledningen till detta ansågs vara att de utförde noggranna tester av de nya systemlösningarna. De tyckte även att det hade varit viktigt att förse användarna med information om hur de skulle komma att påverkas av migrationen. Utbildningssessioner anordnades också efter utförd migration för att användarna skulle kunna använda de nya systemlösningarna på ett effektivt sätt

(Amant och Still, 2007). Om de två andra fallstudierna i Amant och Stills rapport finns det inte så mycket information.

Deras studier resulterade i en identifikation av vanligt förekommande teman och framgångsfaktorer för att en systemlösningsmigration ska bli så framgångsrik som möjligt. Modellen heter *BRW Migration Model for Desktop OSS* och är indelad i olika steg som enligt Amant och Still (2007) bör efterföljas för att öka chansen att migrationsprojektet når ett önskat resultat.

Steg 1 - Skaffa ett organisatorisk engagemang

Steg ett handlar om att se till så att det finns ett stöd hos den ledningsgrupp i den organisation eller förvaltning som systemlösningsmigrationen ska verkställas i. Utan ledningens stöd kan risken öka för att de anställda inte tar migrationen på allvar, och därmed inte börja använda den nya systemlösningen. Detta då de anställda vet att ledningen inte stöttar migrationen full ut, då behöver de inte heller göra det. Det ligger även i människans natur att vara motståndare till förändringar (Amant och Still, 2007). Det är därför viktigt att det finns personer som kan övertyga ledningsgruppen till att stödja den nya systemmigrationen. Det ska också sammanställas vilka resurser som kommer att behövas för migrationsprojektet. Detta kan inkludera den ekonomiska budgeten som finns inom organisationen samt vilken personalkompetens som existerar för tillfället. Detta ska göras för att se hur mycket pengar det finns att förfoga över samt om det finns tillräcklig personalkompetens för att implementera och underhålla en ny systemlösning. En projektplan bör även finnas, innehållande vad målet är med projektet, tidslinje för projektet samt den uppskattade kostnaden av projektet.

Steg 2 - Skapa användarmedvetenhet

Här handlar det om att se till att alla berörda användare av systemet får ta del av migrationsprocessförfarandet. Detta innebär att se till så att användarna förstår:

- Varför det är nödvändigt att byta systemlösning.
- Vad kan de förvänta sig under migrationstiden.
- Vad som kommer att vara annorlunda i den nya systemlösningen.

Om användaren är med på dessa punkter ökar även chansen för att användarna kommer med värdefull information gällande funktionaliteten med det nya systemet. Detta eftersom det är användarna som använder systemlösningen dagligen och vet vilka funktioner de nyttjar. Bra kommunikationen med användarna resulterar då sannolikt i att risken minskar för att irritation och motståndskraft sprider sig inom förvaltningen.

Steg 3 – Genomför utbildning

För att användarna ska kunna nyttja det nya systemet på ett effektivt sätt är utbildning i det nya systemet en viktig del. Att utbilda användarna i det nya systemet ökar inte bara den allmänna datorvanan utan innebär också ett ökat självförtroende hos användaren. Den bästa utbildningen erhålls genom att låta användaren experimentera med systemet samt att hjälpa varandra. Utförs inte detta ökar risken för att systemlösningen används på ett ineffektivt sätt.

Steg 4 – Gör en noggrann analys

Här säger Amant och Still (2007) att det är analysfasen som utgör den största delen i ett migrationsprojekt och att det är nyckeln till framgång. Detta steg innebär att utföra analyser

inom organisation så att den nya systemlösningen bara implementeras på de ställen där det går att förespråka en ökad effektivitet, eller tillgodoser de krav som finns hos användarna och inom organisationen. Detta för att inte riskera att implementera en systemlösning som innebär en försämrad effekt hos användaren eller organisationen. Analysuppgifterna kan variera från projekt till projekt men det finns vissa uppgifter som alltid ska utföras:

- Granska all existerande mjukvara och hårdvara.
- Identifiera de funktionella kraven.
- Identifiera de tekniska kraven.
- Identifiera organisationens affärsstrategi.
- Identifiera de användare som behöver migrera till ny systemlösning.
- Identifiera möjliga öppna källkodslösningar och välj den som uppfyller tidigare krav.
- Skapa en systemspecifikation och systemdesign.
- Definiera den bästa implementeringsstrategin beroende på migrationsprojektets omfattning (big bang, parallell eller gradvis).

Om inte noggrann analys utförs ökar risken för att ekonomiska resurser och viktiga beslut fattas på vaga grunder, vilket kan leda till ett oönskat resultat.

Steg 5 – Sätt upp och installera systemarkitektur och testdatorer

Innan en systemlösningsmigration utförs är det viktigt att relevanta servrar som till exempel filservrar installeras och konfigureras för att kunna vara till hjälp vid en migration. Detta för att stora mängder data kan behöva sparas undan vid en migrationsprocess. Det kan även göra att fördröjningar uppstår i nätverket, när ny programvara ska installeras på många datorer. Eftersom många datorer kan behöva hämta filer från filservrar kan det därför vara bra att fördela filer på ytterligare filservrar för att minska trycket på enskilda servrar och nätverk. Risken för otillgänglig data och försämrad hastighet i nätverket går därför att minska genom väl förberedd planering vid migrationsförfarandet. Testdatorer bör också upprättas för att i nästa steg kunna testa den tänkta systemlösningen.

Steg 6 – Utför tester

Det är viktigt att testa det nya systemet för att säkerställa att önskad pålitlighet uppnås. Detta görs genom att testa de utvalda produkter som valts ut till den nya systemlösningen. Testning kan göras genom att välja ut några testdatorer där användare får testa en nya systemlösningen. Detta för att se om de kan utföra samma uppgifter som gick att göra innan migrationen. Att utföra testning ökar chansen för att hårdvara, mjukvara och pålitlighetsbrister upptäcks innan systemet distribueras ut till en större mängd användare. Detta resulterar i att sannolikheten minskar för att kritiska fel upptäcks efter utförd systemlösningsmigration hos en stor mängd användare.

Steg 7 – Implementera den öppna källkodslösningen

En implementering av den nya systemlösningen på alla utvalda datorer ska göras först när alla möjliga problem som identifierats i tidigare steg är lösta. Implementeringen ska ske enligt den implementeringsstrategi som ska ha utformats i steg 4. Annars ökar risken för att problem upptäcks vid ett skede då de kan vara kostsamma att åtgärda.

Steg 8 – Gör en uppföljning

Detta steget innebär att hela migreringsprocessen dokumenteras och granskas för att kunna utvärdera vad som gick bra och om det uppstod några komplikationer. Detta steg, säger Amant och Still (2007), hoppas ofta över i migrationsprojekt på grund av det kan vara tidskrävande. Dock kan dokumentationen vara mycket bra att inom förvaltningen om ett nytt migrationsprojekt ska göras i framtiden. Dokumentationen bör innehålla:

- Hur migrationsförfarandet gick till (inklusive utbildning).
- Vilka analyser som gjordes och vad som framkom i dessa.
- Vilka ändringar gjordes i systemarkitekturen.
- Vilka eventuella problem som påträffades vid testningsförfarandet.
- Hur eventuella problem och komplikationer löstes.

Detta minskar sannolikheten för att problem inträffar nästa gång.

Steg 9 – Tillgodose fortlöpande användarsupport och utbildning

Det är viktigt att användarsupport och utbildning av användarna fortsätter även efter det att migrationen är slutförd. Det är vanligt att inte alla problem upptäcks under den testperiod som systemet testas och att användare senare stöter på problem i den nya systemlösningsmiljön. Därför är det viktigt att det finns användarsupport tillgänglig för användarna, vilket minimerar risken för minskad arbetseffektivitet och frustration.

3 Problemformulering

3.1 Syfte

Syftet med den här studien är att ta reda på vad som går att lära av andra migrationsprojekt och på så sätt bidra till ökad kunskap inom området systemlösningmigrationer.

3.2 Frågeställningar

- Vilka risker har identifierats i de utförda migrationsprojekten gällande förvaltningarnas dåvarande proprietära systemlösningar och nuvarande systemlösningar baserad på öppen källkod?
- Kan *BRW Migration Model for Desktop OSS* användas för att minska riskerna vid en migration från proprietär systemlösning till en systemlösning baserad på öppen källkod?

3.3 Motivering

Den största anledningen till varför det är intressant att analysera migrationsprojekt där offentliga förvaltningar valt att migrera från proprietära systemlösningar till systemlösningar baserade på öppen källkod, är att det i dagsläget är en högst aktuell fråga, både nationellt och i övriga världen. Detta för att media nu har börjat leverera positiva besked från offentliga förvaltningar som säger sig ha gjort stora ekonomiska besparingar efter att ha utfört systemlösningmigrationer till öppna källkods lösningar (Joinup, 2012c).

Tidigare examensarbeten inom området har bland annat undersökt vilka fördelar och nackdelar som upplevts efter att en offentlig förvaltning migrerat proprietära kontorspaket till kontorspaket baserade på öppen källkod (Sers och Söderlund, 2009). Detta gjordes av Sers och Söderlund som genomförde en studie på en skola i Örebro. Skolan hade utfört en systemlösningmigration från Microsoft Office till OpenOffice. Studien utfördes genom att Sers och Söderlund intervjuade lärarna på skolan, gjorde enkätundersökningar och kompletterande litteraturstudier. Deras resultat visade att den största fördelen efter utförd migration var att skolan slapp betala licenskostnadsavgifter för deras tidigare proprietära alternativ. De använde sig istället av gratisalternativet OpenOffice, som är baserat på öppen källkod. De nackdelar som lärarna på skolan rapporterade var att de efter migrationen upplevde att de avancerade funktionerna var svårare att använda i OpenOffice än vad de hade förväntat sig. Även vissa kompatibilitetsproblem påträffades gällande de filer som skulle kunna läsas både av det nya kontorspaket OpenOffice och av Microsoft Office, som fortfarande fanns kvar på ett fåtal datorer (Sers och Söderlund, 2009). Ett annat examensarbete har fokuserat på att undersöka hur svenska kommuners attityder är till öppna standarder och systemlösningar baserade på öppen källkod (Hansson och Larsson, 2009). Syftet med examensarbetet var att se om det går att lösa de påstådda problem som finns med att använda proprietära systemlösningar, som till exempel inlåsningsseffekt. För att undersöka detta gjorde de intervjuer med utvalda representanter från fyra svenska kommuner för att se hur de såg på öppen källkod och öppna standarder och om detta skulle kunna vara ett alternativ i deras kommun. Hanssons och Larssons slutsatser var att de fyra kommunerna var positiva till öppna standarder, men att de inte såg några belegg för att införande av öppna standarder per automatik löser de problem som är förknippade med proprietära systemlösningar. Representanterna trodde dock att öppna systemlösningar till

viss del bidrar till att minska några problem som till exempel inlåsningsseffekt (Hansson och Larsson, 2009).

Det existerar alltså olika studier som tar upp frågeställningar gällande specifika systemlösningar och specifika förvaltningsområden. Detta gör att det finns utrymme för ett examensarbete som är mer generellt och som inte fokuserar på specifika systemlösningar. Det här examensarbetet ämnar att göra det genom att utföra analyser av litteratur och fallstudier på avrapporterade migrationsprojekt.

Det förväntade resultatet kommer att bestå av två olika delar. Den första delen kommer att bestå av tre analyser som görs på tre utvalda migrationsprojekt. Analyserna kommer sedan att ligga till grund för en sammanställning av identifierade risker före och efter en migrering till systemlösningar baserade på öppen källkod. Den andra delen ämnar att ge insikt i hur lyckade migrationsprojekt har gått tillväga i sina migrationsprocesser. Detta för att analysera om användandet av *BRW Migration Model for desktop OSS* kan minska riskerna i samband med en systemlösningmigration. Analyserna kommer att bidra dels till ökad kunskap inom området systemlösningmigrationer samt till att ytterligare validera *BRW Migration Model for desktop OSS*, vilket även efterfrågas av Amant och Still (2007). De skriver att de gärna ser att modellen används och valideras på fler migrationsprojekt.

Att få all information samlad i en och samma rapport kommer dessutom att underlätta vid informationssökning om risker och migrationsprocesser. Riskerna kommer att presenteras i tabeller vilket gör det lätt att få en överblick över dem. Detta till skillnad mot att försöka urskilja dem i löpande text i de olika migrationsprojektens rapporter på Joinups hemsida. Detta kan vara till hjälp när ett nytt migrationsprojekt ska startas upp, vilket gör att studien framförallt vänder sig till IT-chefer i offentliga förvaltningar.

Att just offentliga förvaltningar valdes var dels på grund av det är enklare att få tag på information kring risker och migrationsprocesser än om kommersiella företag analyserats. Detta för att kommersiella företag med vinstintressen inte gärna vill delge potentiella problem i offentliga forum. Däremot kan även IT-chefer i kommersiella företag ha nytta av resultaten i den här studien, då stegen i *BRW Migration Model for desktop OSS* även kan användas vid migrationer inom kommersiella företag.

Den andra motiveringen till att analysera just offentliga förvaltningar är att de har ett visst ansvar gentemot medborgarna. Ett ansvar som inkluderar hantering av skattebetalarnas pengar på ett ansvarsfullt sätt (Statskontoret, 2003). Statskontoret (2003) menar att informationen hos offentliga förvaltningar många gånger ägs av medborgarna och att de därför har ett ansvar gentemot medborgarna som inkluderar att:

- Garantera medborgarna fri tillgång till offentlig information.
- Upprätthålla bevarandet av offentlig information.
- Garantera säker hantering av offentlig och av medborgarna tillhandahållen information.
- Undvika onödiga utgifter.

Dessa ansvarspunkter gör att det finns ett intresse i att analysera de utförda migrationsprojektens olika systemlösningar före migration och efter migration. Detta för att se om det finns några hot mot de ansvarspunkter som en förvaltning, enligt Statskontoret (2003), bör efterfölja.

3.4 Avgränsningar

Analysen görs på migrationsprojekt som har funnits att tillgå på hemsidan <https://joinup.ec.europa.eu/> under tidsperioden januari-maj år 2012, vilken är den tidsperiod som examensarbetet har utförts under. Analyserade migrationsprojekt har enbart varit migrationsprojekt där en proprietär systemlösning migration till systemlösning baserade på öppen källkod varit aktuell. Slutsatser och resultat görs efter den information som dokumenterats på Europakommissionens samarbetsplattform Joinup (2012c). Det kan vara möjligt att det finns information i ytterligare dokument hos de olika förvaltningarna som inte tas upp i migrationsprojektens rapporter och som skulle kunna påverka resultatet och slutsatserna i den här studien.

Vad gäller de identifierade riskerna i migrationsprojekten skulle det vara möjligt att presentera ytterligare risker i migrationsprojekten. Detta skulle kunna göras genom att anta att andra risker kan uppstå i framtiden med respektive systemlösning. Riskerna som presenteras i resultatet är risker som de tre migrationsprojekten själva identifierat i sina migrationsprojekt och som de på ett eller annat sätt har upplevt. Avgränsningen till detta är för att inte basera resultaten på spekulationer. Det viktiga är att fokusera på risker och information som faktiskt går att styrka i migrationsprojektens rapporter.

De legala aspekterna kring licensiering och upphandlingar i offentliga förvaltningar tas inte upp i den här studien, då fokus inte är på rättsliga frågor.

4 Metod

4.1 Genomförande

För att få en uppfattning om vilka migrationsprojekt som har utförts i offentliga förvaltningar, analyserades först avrapporterade migrationsprojekt som finns publicerade på Europakommissionens samarbetsplattform Joinup (2012c). Plattformen är skapad för att hjälpa publika organ runt om i världen att dela värdefull information kring olika migrationsprojekt baserade på öppen källkod. Joinup är resultatet av en sammanslagning av de tidigare hemsidorna osor.eu och semic.eu. Dessa två hemsidor var före sammanslagningen två åtskilda plattformar, där både information och öppen källkodsmjukvara som använts i tidigare utförda migrationsprojekt fanns tillgängliga för nedladdning. Med den nya plattformen Joinup finns istället all information och mjukvara från de båda hemsidorna samlade på ett och samma ställe, vilket ska förenkla för de som vill ta del av informationen. När den här studien startades, i februari år 2012, fanns det 77 projekt tillgängliga på Joinup. Av dessa 77 projekt var 29 stycken av dem migrationsprojekt som innehöll en migration från proprietär systemlösning till systemlösning baserad på öppen källkod. Resterande projekt fokuserade bland annat på att delge information om olika öppna källkodsmjukvaror utvecklats till specifika ändamål, vilket gör att de inte är av intresse för den här studien.

De urvalskriterier som användes för att välja bland de 29 migrationsprojekten är följande:

- Skrivbordsdatorer

Motivering: Huvudsyftet med migrationsprojektet ska vara migration av skrivbordsdatorer. För att kunna göra jämförelser mellan de olika projekten bör de vara av samma karaktär. Flera projekt på Joinups hemsida inriktade sig på att enbart migrera servrar, men eftersom förutsättningarna blir annorlunda för sådana projekt valdes de bort. Detta kriteriet har även satts upp för att kunna utföra en analys av migrationsprocessen enligt *BRW Migration Model for desktop OSS* (Amant och Still, 2007), då modellen inriktar sig på just skrivbordsdatorer.

- Storlek

Motivering: I urvalet har även storlek på projektet spelat roll. Med storlek avses antal datorer som har varit aktuella för migration. För att öka chansen att hitta en väldokumenterad migrationsprocess väljs de projekt med störst antal datorer ut.

- Avrapporterad del

Motivering: Projektet måste innehålla en signifikant avrapporterad del. Detta för att det ska vara möjligt att kunna sammanställa risker inte bara före, utan också efter migrationen. En signifikant avrapporterad del krävs också för att kunna följa de olika stegen i *BRW Migration Model for desktop OSS* (Amant och Still, 2007).

- Aktuellt

Motivering: För att resultaten ska ha relevans så prioriterades projekt som var så nya som möjligt. Detta för att tekniken utvecklas i snabb takt och snabbt blir gammal.

Efter att ha sammanställt projekten enligt urvalskriterierna valdes de tre migrationsprojekt som uppfyllde kriterierna bäst ut för analys. Det var Linux-projektet i Münchens stad,

franska gendarmeriets migrationsprojekt och skolprojektet i Bolzanoprovinsen. Det fanns fler projekt som också var stora i omfattning eller som var relativt aktuella tidsmässigt, men de matchade inte alla kriterierna. Finska justitieministeriets projekt var ett av de större, sett till antal datorer. Deras migrationsprojekt startade dock år 2007 och rapporten på Joinup skrevs redan i februari år 2007, vilket innebär att deras rapport inte innehöll tillräckligt med information för att göra en analys av riskerna efter en migration. Svenska rikspolisstyrelsens projekt var relativt aktuellt, men deras projekt gällde endast servrar och inte skrivbordsdatorer. För att se en komplett lista över vilka migrationsprojekt som fanns i urvalet, se appendix A.

4.2 Fallstudie

Fallstudiemetoden används i den här studien för att analysera avrapporterade migrationsprojekt. Beslutet om att använda fallstudie som metod stärks av hur Merriam (1994) definierar en fallstudie i boken "Fallstudien som forskningsmetod". Där säger Merriam (1994) att en fallstudie är partikularistisk, vilket betyder att fallstudien består av en undersökning av en specifik företeelse. Detta stämmer också in på vad den här studien ämnar göra, det vill säga att analysera systemlösningsmigrationer i offentliga förvaltningar som migrerat en proprietär systemlösning till en systemlösning baserad på öppen källkod. Fallstudien är också induktiv, vilket Merriam (1994) beskriver som att fallstudien grundar sig på ett induktivt resonemang, det vill säga att de begrepp och slutsatser som görs i resultatet härleds från erfarenheter.

4.3 Litteraturstudie

Berndtsson m.fl. (2008) beskriver att en litteraturstudie är en metod där publicerade källor systematiskt analyseras med ett specifikt problem eller mål i åtanke. Den här metoden har därför använts i rapportens bakgrundskapitel. Detta för att systematiskt analysera publicerad litteratur gällande definitioner som är kopplade till risker och systemlösningar. Berndtsson m.fl. (2008) säger att det finns olika tekniker som kan användas för att hitta relevanta källor för litteraturstudien. En teknik är att använda bibliografiska databaser för att söka på relevanta nyckelord som skulle kunna matcha det identifierade problemet. När en referens är hittad går det att få fram ytterligare referenser genom dennes referenslista, som kan vara till hjälp för att hitta ytterligare information. En annan teknik som beskrivs är att söka på internetsidor bland vetenskapliga journaler för att hitta relevant information.

Berndtsson m.fl. (2008) tar också upp ett problem med litteraturanalyser som är viktigt att ha i åtanke när en litteraturstudie genomförs. Det är att veta när insamlat material är tillräckligt för att uppnå validitet på studien. Berndtsson m.fl. (2008) säger också att det är svårt att veta exakt vad som krävs för uppnå den önskade validiteten på studien. Om processen förklaras för läsaren och det argumenteras för hur resultaten har växt fram genom litteratursökningar ökar chansen för ett trovärdigt resultat.

4.4 Validitet och reliabilitet

Enligt Berndtsson m.fl. (2008) är validitet förhållandet mellan det som ämnas undersökas och det som faktiskt undersöks. För att skapa en så bra validitet som möjligt i studien gäller det att vara medveten om vilka hot det finns mot validiteten. Berndtsson m.fl. (2008) säger att ett möjligt hot är författarens oförmåga att redovisa för sin personliga åsikt. Detta betyder att författaren med stor sannolikhet har en förutfattad mening när denne tar sig an en uppgift eller en studie. Det är då möjligt att dessa förutfattade meningar kommer att påverka hur arbetsprocessen kommer att se ut och vad resultatet kommer att visa. Med detta i åtanke

har analyserna granskats på ett så objektivt sätt som möjligt. Avslutande reflektioner över det egna arbetet redovisas också under kapitlet "Diskussion".

Reliabilitet beskriver Berndtsson m.fl. (2008) som ett sätt att mäta hur tillförlitlig metoden av studien är. Det vill säga att det finns ett systematiskt angreppssätt på hur informationsinsamlandets förfarande och resultat är utformat. Merriam, (1994) säger att reliabilitet handlar om i vilken utsträckning resultat går att upprepa. Alltså om undersökningen/analysen kommer att ge samma resultat om den upprepas, oavsett vem som gör det. För att skapa en större reliabilitet i den här studien användes referensmaterial och övrig litteratur från vetenskapligt erkända sökmotorer och databaser. De mest frekventa databaser som använts är ACM Portal, IEEE Explore och ScienceDirect. Informationssökning har skett direkt via respektive databas eller via sökmotorn Google Scholar. Detta medför att om någon skulle vilja återskapa studien, kan samma källor användas för att reproducera så mycket som möjligt av resultaten.

4.5 Alternativ metod

En alternativ metod skulle kunna vara att göra ett experimentellt test. Berndtsson m.fl. (2008) säger att den experimentella metoden kan användas för att testa en given hypotes. Detta med syftet att se om det går stödja hypotesen eller inte. Dock säger Berndtsson m.fl. (2008) att resultatet av ett experiment inte kan bevisa en hypotes fullt ut. Detta för att det alltid kan finnas en annan förklaring till varför resultaten blev som de blev. För att undersöka hypotesen om att *BRW Migration Model for desktop OSS* kan användas för att minska riskerna vid en systemmigration hade det varit intressant att pröva den i realtid på ett pågående migrationsprojekt i en offentlig förvaltning. De olika stegen i modellen hade då kunnat analyseras i projektet under tiden det pågick, istället för i efterhand. Det ansågs dock inte vara rimligt i det här examensarbetet på grund av att tidsramen är begränsad och en migrationsprocess ofta genomförs under en längre period.

5 Resultat

De tre migrationsprojekten som har legat till grund för analys och resultat är tre migrationsprojekt som utförts i Tyskland, Frankrike och Italien. Projektet i Tyskland, som kallas LiMux-projektet, består av att Münchens stad bestämde sig för att de ville migrera cirka 14.000 arbetsdatorer till att använda systemlösningar baserade på öppen källkod. Innan migreringen användes främst proprietära systemlösningar, som till exempel operativsystemet Microsoft Windows. Efter utförd migration används bland annat operativsystemet Ubuntu och kontorspaketet OpenOffice (Joinup, 2012a).

Det franska gendarmeriet, som är en del av den franska polisen, gör en liknande migration med liknande systemlösningar. Migrationsomfattningen är något större än migrationsprojektet i Münchens stad och involverar cirka 15.000 arbetsdatorer (Joinup, 2012b).

Bolzanoprovinsen är en provins i norra Italien som startade ett migrationsprojekt år 2005, vilket gick under namnet FUSS (Free Upgrade South Tyrol's Schools). Migrationsprojektets omfattning bestod i att 72 skolor i provinsen skulle migrera sina cirka 2500 datorer till systemlösningar baserad på öppen källkod (Joinup, 2012d). För ytterligare information om de tre migrationsprojekten finns en mer detaljerad summering i appendix B, appendix C och appendix D.

Efter att ha analyserat de tre migrationsprojekten består resultatet av en sammanställning av projektens identifierade risker före och efter en partiellt utförd systemlösningsmigration. De identifierade riskerna före en systemlösningsmigration är sammanställda i tabell 1 och de identifierade riskerna efter en systemlösningsmigration är sammanställda i tabell 2 nedan. För att läsa om varför en risk är identifierad och vad risken innebär, följ identifikationsnumret i tabellerna som är kopplad till en förklaring med samma identifikationsnummer.

5.1 Identifierade risker före systemlösningsmigration

En analys av de utvalda migrationsprojekten är gjord för att se vilka risker som förvaltningarna har identifierat före påbörjade systemlösningsmigrationer, det vill säga risker som är förknippade till ett användande av proprietära systemlösningar.

Tabell 1: Identifierade risker före systemlösningsmigration

ID	Risker före	Münchens stad	Franska gendarmeriet	Bolzano-provinsen
1	Inte kunna ta egna IT-beslut	Ja	Ja	Nej
2	Inte ha en modulär systemlösning	Ja	Ja	Ja
3	Betala licensavgifter	Ja	Ja	Ja
4	Köpa ny hårdvara	Framgår ej	Ja	Ja
5	Ej läsbar data över tid	Ja	Framgår ej	Framgår ej

Teckenförklaring: Ja = Risken är identifierad, Nej = Risken är inte identifierad, Framgår ej = Framgår inte av migrationsprojektets rapport.

1. När en förvaltning använder en proprietär systemlösning finns risken att förvaltningen inte kan fatta sina egna IT-beslut. Det betyder att om förvaltningen i framtiden vill uppgradera eller skapa någon extra funktionalitet i systemlösningen är det sannolikt att det enda alternativet för att åstadkomma detta är genom den valda leverantörens alternativ (Simon, 2005). Det kan då innebära att en uppgradering av systemlösningen mer eller mindre blir påtvingad. Detta kan ske på grund av att leverantören till exempel väljer att avbryta supporten för aktuell systemlösning, vilket innebär att säkerhetsuppdateringar slutar distribueras ut till systemlösningen. Detta var en risk som gick att identifiera både i Münchens stad och det franska gendarmeriet. Bolzanoprovinsen nämner inte att de upplevde någon risk för att inte kunna ta sina egna IT-beslut.
2. Att använda en proprietär systemlösning kan innebära en ökad risk för att systemlösningen inte är modulär, vilket kan leda till att tilläggsmoduler och påbyggd funktionalitet kan bli svår att åstadkomma. Detta har upplevts i de tre avrapporterade migrationsprojekt, eftersom de använt sig av en systemlösning som inte fick lov att modifieras av användaren, enligt det EULA-avtal (Edwards, 2005) som medföljer och reglerar de systemlösningar som använts.
3. När en offentlig förvaltning använder sig av proprietära systemlösningar ökar risken för att pengar används till att betala programvarulicenser. Det är pengar som skulle kunna användas till andra saker inom förvaltningen, om en gratis systemlösning baserad på öppen källkod istället skulle användas. Alla tre migrationsprojekt har före utförd migration spenderat ekonomiska resurser på att betala licensavgifter för de proprietära systemlösningar som de har använt.
4. Det franska gendarmeriet upplevde risken att behöva lägga pengar på att köpa in nyare och bättre hårdvara till ett stort antal datorer. Detta hade varit nödvändigt om det franska gendarmeriet hade valt det uppgraderingsalternativ som Microsoft erbjöd som ersättning för den systemlösning där Microsofts support skulle upphöra. Uppgraderingsalternativet hade mycket hårdare systemkrav, vilket hade krävt ny hårdvara (Joinup, 2012b). Detta kan resultera i en stor ekonomisk utgift beroende på förvaltningsstorlek. Bolzanoprovinsen säger att de med den nya systemlösningen inte behöver uppgradera sin hårdvara i den takt som tidigare hade behövts med det proprietära alternativet. Detta för att operativsystem baserade på Linux inte kräver lika mycket prestanda som till exempel operativsystemet Microsoft Windows (Joinup, 2012d). Det framgår inte om projektet i München upplevde risken för att behöva köpa in ny hårdvara om en migration inte hade påbörjats.
5. Risken för att inte kunna läsa lagrad data i stängda filformat efter en längre tidsperiod ökar om en förvaltning använder en proprietär systemlösning. Om till exempel en systemlösning ska bytas ut eller försvinna kan det finnas en risk för att det inte går att läsa/öppna gamla filer med den nya systemlösningen (Joinup, 2012a). Detta är en risk som hotar den ansvarspunkt som enligt Statskontoret (2003) innebär att en offentlig förvaltning har ett ansvar gentemot medborgarna att bevara offentlig information. Den här risken upplevdes som stor i Münchens stad på grund av att de använde proprietära systemlösningar. Risken diskuterades inte i de andra migrationsprojekten.

5.2 Identifierade risker efter systemlösningsmigration

De identifierade riskerna efter utförd systemlösningsmigration innebär risker som förvaltningarna har identifierat efter att de har migrerat till systemlösningar baserad på öppen källkod. En sammanställning av detta presenteras i tabell 2 nedan.

Tabell 2: Identifierade risker efter systemlösningsmigration

ID	Risker efter	Münchens stad	Franska gendarmeriet	Bolzano-provinsen
1	Behov av extern support	Ja	Ja	Ja
2	Uppleva vissa kompatibilitetsproblem	Ja	Nej	Nej
3	Utbildningskostnader	Ja	Ja	Ja
4	Ej uppnå samma funktionalitet	Ja	Nej	Ja

Teckenförklaring: Ja = Risken är identifierad, Nej = Risken är inte identifierad, Framgår ej = Framgår inte av migrationsprojektets rapport.

1. Bara för att en systemlösning är baserad på öppen källkod innebär det inte att extern support inte är nödvändig. Förvaltningarna kan fortfarande vara beroende av distributörer av specifika systemlösningar när det gäller uppdateringar och support (Ven m.fl. 2008). Münchens stad anlidade externa företag för att få hjälp med rådgivning och programmeringsfrågor. De räknar att det mesta av detta ska skötas internt när migrationen är helt slutförd (Joinup, 2012a). Det franska gendarmeriet valde att anlita en kommersiell aktör för konsultation när det gällde komplexa frågor i framtiden med den nya systemlösningen. Detta skulle kunna vara en potentiell risk eftersom de riskerar att framtida beslut baseras på utomstående kommersiell aktör. Den kommersiella aktören skulle kunna rekommendera sådant som inte ser till förvaltningens bästa, utan gynnar aktören själv ekonomiskt. Bolzanoprovinsen anlidade ett externt företag innan migration för att ta hand om de tekniska migrationsdetaljerna. Efter utförd migration visade det sig att det fanns en saknad för ett grafiskt gränssnitt för lärarna gällande användarhantering i systemet. Detta kunde inte lösas internt och samma företag jobbar nu med att utveckla ett sådant grafiskt gränssnitt (Joinup, 2012d).
2. Münchens stad har fortfarande kvar några proprietära systemlösningar som inte ansetts vara lönsamma att migrera eller där en migration inte upplevts vara teknisk möjlig. Detta har gjort att det uppstått vissa kompatibilitetsproblem, till exempel när ett dokument ska användas av både Microsoft Office och OpenOffice (Joinup, 2012a). För att minimera kompatibilitetsproblemen har de satt upp riktlinjer för hur ett dokument ska utformas för att kunna hanteras av båda kontorspaketen. Det franska gendarmeriet och Bolzanoprovinsen säger att de inte upplevde några direkta kompatibilitetsproblem kopplade till de nya systemlösningarna baserade på öppen källkod.
3. Münchens stad såg möjligheten att öka datakompetensen hos användarna i samband med den aktuella systemlösningsmigrationen. De ansåg dock inte att behovet av utbildning var kopplat direkt till att systemlösningen var baserad på öppen källkod.

De menade att många användare redan innan inte kunde hantera den tidigare proprietära systemlösningen på ett effektivt sätt (Joinup, 2012a). De systemadministratörer som skulle underhålla de nya systemlösningarna gavs utbildning i underhåll och de övriga anställda gavs utbildning för att kunna hantera den nya systemlösningen så effektivt som möjligt i sitt arbete. Om en siffra ska sättas på hur mycket upplärningskostnaden har varit för projektet, säger projektgruppen att om de räknar med den tid som ägnats åt upplärning och baserar det på deras femårsplan så hamnar kostnaden på cirka 13.3 miljoner euro. Detta innebär då 38% av migrationsprojektets totala kostnad (Joinup, 2012a). Det franska gendarmeriet säger sig inte ha upplevt så stora utbildningskostnader då de tyckt att deras systemlösning varit så pass användarvänlig. Dock skapades en applikation med vanligt förekommande frågor kring systemlösningarna. En telefonlinje upprättades också vid akuta problem. Det finns inga siffror på dessa utgifter i rapporten (Joinup, 2012b). Bolzanoprovinsen genomförde utbildningar för att lärarna på skolorna skulle kunna använda systemlösningarna och kunna hjälpa studenterna använda dem på ett fullgott sätt. Detta kostade cirka 70.000 euro att genomföra.

4. Användarna av den nya systemlösningen i München upplevde att de inte kunde hantera den funktion som hanterade dokumentmallar och andra kopplade dokument i den nya systemlösningen. Projektteamet skapade därför en modul bestående av ett verktygsfält som liknade den tidigare systemlösningen. Detta ska dock utredas för att se om det är en bra lösning (Joinup, 2012a). Bolzanoprovinsen rapporterade att de upplevt att den nya motsvarigheten till deras dåvarande designmjukvara inte har lika stor funktionalitet som innan. Dock säger de att de inte anser att studenterna gör så pass avancerade designprojekt så de flesta funktioner kommer antagligen inte att saknas. De lärare som ska ha rättigheter till att administrera användare i systemlösningarna saknade också ett grafiskt gränssnitt för att utföra detta. Problemet är dock på väg att lösas genom en påbörjad utveckling av ett grafiskt gränssnitt i den nya systemlösningen (Joinup, 2012d). Det franska gendarmeriet säger sig inte ha upplevt någon försämrad funktionalitet med den nya systemlösningen.

5.3 Analys efter migrationsprocessmodell

För att undersöka om *BRW Migration Model for desktop OSS* kan användas för att minska riskerna vid en migrationsprocess presenteras här analyser av de tre migrationsprojekten. Migrationsprocesserna för de tre projekten har analyserats efter de nio stegen i migrationsprocessmodellen, se avsnitt 2.3. Inget av de tre utvalda migrationsprojekten har använt sig av den föreslagna migrationsprocessmodellen när de har utfört sina migrationsprocesser. Analyserna nedan baseras på migrationsprojektens avrapporteringar på Joinups hemsida (Joinup, 2012c).

5.3.1 Münchens stad

Sammanfattning: LiMux-projektet i Münchens stad har haft ett angreppssätt som har varit förhållandevis likt de föreslagna migrationsförfarandet som *BRW migration model for desktop OSS* anvisar. Detta skulle kunna vara anledningen till att inte några större problem har inträffat inom migrationsprojektet. Enligt projektledningen har den främsta anledningen till önskvärt resultat varit att berörda parter varit överens innan en migrationsprocess påbörjats. Detta kan relateras till vad migrationsprocessmodellen anser är viktigt att fokusera på, nämligen att det är viktigt att hela förvaltningen är delaktig och har en positiv inställning

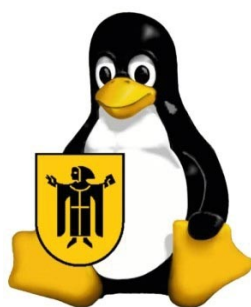
till en systemlösningsmigration (Amant och Still, 2007). Nedan presenteras i detalj hur migrationsprojektet gått tillväga i migrationsprocessen.

Steg 1 – Skaffa ett organisatorisk engagemang

Idén om att kunna fatta sina egna IT-strategiska beslut väcktes hos stadsfullmäktige i Münchens stad och hade alltså redan från början högsta ledningens stöd för att påbörja en migration. Münchens stad kontaktade en konsultfirma som fick i uppdrag att studera olika systemlösningar för att hitta den som var bäst lämpad utifrån deras behov och krav. Resultatet blev två kostnadsförslag som inkluderade en lösning med proprietära systemlösningar och ett förslag med systemlösningar baserade på öppen källkod. Båda kostnadsförslagen innehöll information som inkluderade alla migrationskostnader förknippade med projektet, inklusive personalkostnader och utbildningskostnader.

Steg 2 – Skapa användarmedvetenhet

De projektansvariga lade ner mycket resurser på att övertyga samt skapa ett organisatorisk engagemang hos hela förvaltningen. De använde sig av olika prestandademonstrationer och förevisningar för att låta användarna testa kommande systemlösningar. Ytterligare kommunikationsmedel som e-post, intranät, nyhetsbrev, forum och wikis användes för att förmedla projektets framsteg. De anställda gavs även möjligheter att komma med förslag och andra synpunkter på hur de olika systemlösningarna kunde förbättras. För att ytterligare öka intresset hos användarna skapade de också en maskot som skulle förknippas med projektet. Maskoten fick namnet Mux och var en modifiering av Linux officiella maskot Tux.



Figur 1: Mux

(Hämtad på: http://www.muylinux.com/wp-content/uploads/2012/03/limux_logo.jpg)

Steg 3 – Genomför utbildning

Här valde Münchens stad att försöka öka användarnas kompetens genom att satsa på grundläggande samt specialriktad upplärning. Detta innebar att all personal fick ta del av en grundlig utbildning i de vanligaste funktionerna som de behövde kunna i den nya systemlösningen. Efter det fick användarna välja vilken typ av vidareutbildning de behövde. Detta gestaltades i olika moduler som innehöll specifika utbildningar beroende på vad användarna behövde kunna i sitt dagliga arbete. När det gällde systemadministratörerna fick dessa först en fyra dagars utbildning i hur de skulle installera och distribuera de nya systemlösningarna.

Steg 4 – Gör en noggrann analys

Förvaltningen analyserade vad de hade för infrastruktur och vad deras krav och behov var hos den systemlösning de ämnade implementera. Analysen resulterade i att den nya systemlösningen inte skulle implementeras på alla ställen i förvaltningen. Proprietär programvara skulle användas antingen där en lösning med öppen källkod är teknisk omöjlig,

eller där den inte är kostnadseffektiv gentemot förvaltningen. Detta gällde framförallt de 300 administrativa mjukvarupaket som identifierats inom förvaltningen.

Lösningförslaget till dessa mjukvarupaket resulterade i att Münchens stad ville att dessa åtminstone skulle bli plattformsoberoende och att de skulle kunna köras via en webbläsare. Om detta ej var möjligt fanns tre alternativ:

- Köra programmet i en virtuell Windowsdator på en Linux klient.
- Använda Wine-emulator (Winehq, 2012) för att skapa en fungerande exekveringsmiljö.
- Köra programmet på en terminalserver.

Om inget av dessa alternativ skulle vara tekniskt möjliga att implementera skulle programmet köras på en Windowsdator. Den här analysfasen resulterade även i att ett beslut om att en gradvis migrationsprocess skulle genomföras. Detta innebar att inte hela migrationsförfarandet gjordes på en gång likt en big bang-migration, vilket betyder att en migration utförs genom att stoppa all aktivitet och sedan byta ut hela infrastrukturen på en och samma gång.

Steg 5 – Sätt upp och installera systemarkitektur och testdatorer

Detta steg innebar inte så mycket för förvaltningen eftersom de kunde använda redan existerande systemarkitektur. Enligt BRW-modellen ska i detta steg till exempel nya filservrar köpas in för att hantera all data som måste sparas undan. Detta gäller dock främst förvaltningar som gör sin migration av typen big bang. Förvaltningen hade redan valt att göra en gradvis migrering och behövde därmed inte hantera lika stora datavolymer som en migration av typen big bang skulle inneburit.

Steg 6 – Utför tester

Resultaten av att testa systemlösningar visade att vissa kompatibilitetsproblem upptäcktes mellan de 13.700 dokumentmallar som tidigare använts till dåvarande kontorspaketet Microsoft Office. Många av dessa var nu oanvändbara i det nya kontorspaketet OpenOffice. Tester ledde bland annat fram till att riktlinjer för hur en dokumentmall skall utformas för att få så lite kompatibilitetsproblem som möjligt. De skapade också en ny plugin till OpenOffice (Wollmux) som gjorde att dokumentmallar nu kunde skötas centralt av IT-administrationen, vilket inte var möjligt innan.

Tester har även utförts för att se vilka av de existerande samt nya programvarorna som kan köras på den nya lösningen baserad på öppen källkod. För de som inte klarade kraven har en annan lösning fått implementeras (se Steg 4).

Steg 7 – Implementera den öppna källkodslösningen

När implementeringsfasen inleddes gjordes det enligt det gradvisa migrationsförfarandet som identifierats i steg 4. Migrationsprocessen startade med att först byta ut de mest använda programmen inom förvaltningarna till de likvärdiga alternativen Firefox, OpenOffice och Thunderbird. Dessa installerades på användarnas dåvarande Windowsdatorer för att inte skapa ett för stort miljöombyte för användarna. Om inte den här gradvisa migrationen valts hade det kunnat resultera i att användarna av systemen hade fått en mycket brant upplärningsperiod med att läras sig ett nytt operativsystem tillsammans med en mängd nya programvaror. Projektkoordinatorn Schießl säger också att den gradvisa

migrationen gjorde att utbildningsförfarandet gick att hålla på en rimlig nivå. När användarna på de olika avdelningarna vant sig vid de nya programmen och de kände sig redo, kunde de välja att även byt till det specialframtagna operativsystemet LiMux.

Steg 8 – Gör en uppföljning

Projektets genomförande har dokumenterats i de flesta avseende som *BRW migration modell* föreskriver att en implementeringsåterblick bör innehålla. Upplärningsförloppet har beskrivits beträffande hur de har gått till väga och vad deras mål har varit (se steg 3). De har beskrivit de analyser som har gjorts i migrationsprojektet. Det vill säga kostnadsförslag, teknisk interoperabilitet, utbildning och detaljer i migrationsförfarandet. Påträffade problem har dokumenterats och lösningsförslagen till dessa är förklarade. Detta gäller de mest uppmärksammade komplikationer som hittills har redogjorts för i aktuellt migrationsprojekt som funnits att tillgå för den tidsperiod den här analysen utförts.

Steg 9 – Tillgodose fortlöpande användarsupport och utbildning

För att användarna ska kunna ta del av den utbildning och träning som de alla har gjort i samband med migrationen, skapade projektteamet e-lärnings plattformen *Limux Lernwelt*. Detta gjorde att användarna kunde återupprepa de utbildningslektioner de tidigare fått, när det fanns behov av repetition. Användarna kunde ansluta sig till plattformen från sina egna arbetsdatorer. Detta innebar att de kunde repetera lektionerna i sin egen inläringstakt.

5.3.2 Franska gendarmeriet

Sammanfattning: Det franska gendarmeriet har inte avrapporterat en lika detaljerad rapport som Münchens stad. Detta har bidragit till att information om tillvägagångssättet i vissa steg av migrationsprocessmodellen ej har gått att validera fullt ut. Dock går det att se från informationen i rapporten att det franska gendarmeriet har haft en lyckosam migrationsprocess i de steg som anses vara av stor vikt i migrationsprocessmodellen. Detta betyder att de har haft fullt stöd från ledningsgruppen och de har även noga utfört analyser för den omgivning som systemlösningsmigrationen kommer att påverka. Nedan följer mer i detalj hur det franska gendarmeriet gått tillväga i sin migrationsprocess.

Steg 1 - Skaffa ett organisatorisk engagemang

På grund av att franska gendarmeriet redan år 2001 tagit beslutet att deras systemlösningar skulle behöva bli mer modulära, fanns det redan ett etablerat engagemang hos ledningsgruppen. Den grupp av IT-experters som var ansvariga för migrationsprocessen kom snabbt fram till att öppen källkod var vad som skulle möta deras modulära behov. Det var ingen som motsatte sig beslutet och en migrationsprocess kunde därför påbörjas så snabbt som möjligt.

Steg 2 - Skapa användarmedvetenhet

I rapporten framgår det inte om de gjorde något för att skapa en användarmedvetenhet eller om migrationen utfördes utan att informera användarna varför och vad de kunde komma att förvänta sig.

Steg 3 – Genomför utbildning

Det franska gendarmeriet rapporterar att de upplevde oväntat lite behov av utbildning av användarna gällande operativsystemmigrationen av Microsoft Windows till Ubuntu. De säger att den främsta anledningen till det är att Ubuntu är väldigt användarvänligt. De tror till

exempel att en migrering från dåvarande Windows XP till Windows Vista skulle varit svårare för användarna att hantera, då det hade inneburit en större skillnad i design och funktionalitet än vad en migrering till Ubuntu innebar.

Steg 4 – Gör en noggrann analys

De som var ansvariga för att planera migrationsprocessen utgick från fyra frågeställningar för vad som skulle ingå på en normal arbetsdator. Frågeställningarna var följande:

- Vad behöver vi?
- Vilka applikationer är beroende av Microsoft?
- Vilka av dessa är kompatibla med Ubuntu?
- Var är de aktuella arbetsdatorerna utplacerade?

Resultatet av att analysera de fyra frågeställningarna blev en lista på 35 applikationer som ansågs vara nödvändiga för en normal arbetsdator. Om de hittade en fungerande lösning till alla dessa framgår inte i rapporten.

Steg 5 – Sätt upp och installera systemarkitektur och testdatorer

Det framgår inte om någon systemarkitektur installerades för att undvika ett för stort tryck på filserverar och nätverksutrustning.

Steg 6 – Utför tester

Inte heller här framgår om en testningsprocess har funnits. Dock var många av systemadministratörerna redan bekanta med operativsystem baserade på Linux. Detta eftersom deras serverar redan använde Debian som operativsystem, vilket kan ha medfört att tester inte behövde utföras i lika stor utsträckning.

Steg 7 – Implementera den öppna källkodslösningen

Implementeringsmetoden som användes var en stegvis migrationsprocess. De migrerade först webbläsare och kontorspaket på deras dåvarande Windowsdatorer. Sedan migrerade de operativsystemen till Ubuntu, vilket medförde en mjukare övergång för användarna.

Steg 8 – Gör en uppföljning

Den dokumentation som finns tillgänglig på Joinup (Joinup, 2012c) är ganska generell. Det finns lite information om de punkter som enligt migrationsprocessmodellen bör ingå i en implementeringsåterblick i det avrapporterade migrationsprojektet. Om det finns en mer detaljerad dokumentation kring den utförda migrationsprocessen framgår ej.

Steg 9 – Tillgodose fortlöpande användarsupport och utbildning

På varje arbetsdatorer installerades ett handledningsverktyg som skulle hjälpa användarna med problem som skulle kunna uppstå. De inrättade även en telefonlinje som användarna kunde använda för att få handledning gällande mer tekniska frågor. Dock säger de att telefonlinjen inte används så ofta och de har märkt att de flesta användarna lärt sig genom att själva sitta och utforska programvarorna.

5.3.3 Bolzanoprovinsen

Sammanfattning: Resultatet av den utförda systemlösningsmigrationen i Bolzanoprovinsen har enligt de involverade varit positivt. Systemlösningsmigrationen har utförts på cirka 2500 datorer som är utplacerade på 72 skolor i provinsen. De har nu en IT-miljö baserad på öppen källkod, som gör att de slipper att betala licensavgifter till kommersiella företag. En sak som de hade kunnat göra bättre var att de skulle ha involverat lärarna mer i utvecklingsfasen av den specialanpassade Linux-distributionen FUSS Soledad. De säger dock att den viktigaste delen i migrationsprocessen var den noga utförda analysfasen som gjordes. Även fast analysfasen krävde mycket arbete tror de att den var en viktig del i projektets succé. Det går alltså att dra tydliga kopplingar även i detta projektet mellan vad migrationsprocessmodellen tycker är viktigt att fokusera på och vad som upplevts som viktiga faktorer i detta migrationsprojektet. Den viktigaste faktorn har alltså varit att utföra noggranna analyser för att inte systemlösningar implementeras på ställen där till exempel funktionalitet eller tekniska krav efteråt ej visa sig vara tillräckliga.

Steg 1 - Skaffa ett organisatorisk engagemang

Idén om att migrera de proprietära systemlösningarna till öppna källkodslösningar uppkom år 2004. Datorerna på skolorna i Bolzanoprovinsen behövde uppgradera sina systemlösningar och när en IT-lärare på en av skolorna pratade med en representant från skolmyndigheten i Bolzanoprovinsen visade det sig att de båda delade samma vision om en öppnare IT-miljö. De startade därför tillsammans ett migrationsprojekt som gick under namnet FLUSS (Free Upgrade South Tyrol's Schools). I och med att det fanns stöd både i lokala skolor och i skolmyndigheten kunde ett beslut snart fattas om en påbörjad migrationsprocess.

Steg 2 - Skapa användarmedvetenhet

Projektgruppen valde att inte gå ut i media med nyheten om att de skulle migrera skolornas systemlösningar till öppen källkod. Det var egentligen bara lärarna som meddelades om att ett migrationsförfarande skulle ske under sommarlovet 2005. Projektgruppen menade att de inte ville gå ut med informationen till media då de ansåg att det skulle medföra stor press på att de skulle behöva lyckas. Efter utförd migration har det dock visat sig att vissa lärare inte känt att de varit tillräckligt delaktiga i utvecklandet av hur den nya Linux-distributionen skulle utformas.

Steg 3 – Genomför utbildning

För att alla studenter skulle kunna bekanta sig med den nya systemlösningen skapades 20.000 LiveCDs som innehöll det nya operativsystemet med tillhörande programvaror. Dessa LiveCDs skickades sedan ut till de berörda familjerna så att de skulle kunna träna sig på att hantera de nya systemlösningarna. För att lärarna skulle kunna hantera systemlösningarna på ett tillfredsställande sätt och därmed kunna vara till hjälp för eleverna om de stötte på problem i den nya IT-miljön, anordnades två olika kurser. Den första kursen var en grundkurs på 20 timmar. Här fick lärarna lära sig att använda den nya skrivbordsmiljön Gnome och det nya kontorspaketet OpenOffice. Utbildningen utfördes med de datorer som lärarna skulle använda i sina egna klassrum. Den andra kursen var avsedd för de 80 lärare som skulle ansvara för de nya IT-systemen på respektive skola och var mer tekniskt avancerad. Den pågick därför i 40 timmar på Einaudis yrkesskola. Om det skulle uppstå problem som inte kunde lösas lokalt på respektive skola utbildades även en grupp på åtta personer för att kunna hjälpa de olika skolorna om speciella problem skulle uppstå. Dessa fick cirka 600 utbildningstimmar och var nu klassade som systemadministratörer.

Steg 4 – Gör en noggrann analys

Innan genomförandet, i februari år 2005, skickades det ut frågeformulär till de 80 lärarna som var ansvariga för IT-systemen på skolorna. Frågeformulärets funktion var att ta reda på vilken hårdvara och mjukvara som användes på respektive skola. Alla 80 lärare intervjuades också om hur just deras skola använde sig av de olika datorerna och systemlösningarna. De tillfrågades även om vilka systemlösningar de skulle vilja ha i den nya systemlösningen. I projektets rapport framgår det att resultatet av de utförda analyserna och intervjuerna som gjordes med lärarna var till stor hjälp vid utvecklingen av den nya Linux-distributionen.

Steg 5 – Sätt upp och installera systemarkitektur och testdatorer

Av den information som finns tillgänglig om detta projekt framgår det inte om de installerade någon extra systemarkitektur eller testdatorer. Dock står det att de bad alla skolor att spara undan alla data som behövde finnas kvar innan påbörjad migration.

Steg 6 – Utför tester

Av informationen som finns i rapporten framgår det inte om några testdatorer upprättades. Den nya systemlösningen implementerades direkt efter skapandet av FUSS Soledad som var baserad på utförda behovsintervjuer.

Steg 7 – Implementera den öppna källkodslösningen

Implementeringen skedde under sommarlovet när inga lärare eller studenter fanns på skolorna. Metoden för att implementera de nya systemlösningarna på de cirka 2500 datorerna gjordes likt en big bang-implementering. De migrerade alla datorer som enligt deras analys varit aktuella för migration. De säger också att migrationsförfarandet gick bra och att det faktum att de inte berättade för media om migrationen gjorde att de kunde arbeta under mindre press.

Steg 8 – Gör en uppföljning

En sammanställd dokumentation har gjorts där det står vilka problem som upplevts och hur de har gått tillväga för att genomföra migrationsprocessen. De projektansvariga säger de lyckade resultatet av projektet till stor del beror på de väl utförda analyser som gjordes för att se vilka behov som efterfrågades. Att dessa väl utförda analyser, tillsammans med resterande dokumentation om projektet, finns tillgängliga gör att problem kan undvikas vid framtida migrationer.

Steg 9 – Tillgodose fortlöpande användarsupport och utbildning

För att lärare inte ska behöva känna sig osäkra i den nya IT-miljön har projektgruppen ordnat så att supportpersonal kommer på besök en gång i veckan till varje skola. Syftet är att kunna hjälpa till om problem eller frågor gällande de nya systemlösningar skulle uppstå. Detta har gjort att lärarna har känt en viss säkerhet. De vet att även om problem skulle uppstå så kommer det alltid någon som kan bistå med hjälp minst en gång i veckan.

6 Diskussion

Det har inte varit möjligt att dra några paralleller till de fallstudier som analyserats i den här studien och de som Amant och Still (2007) baserat sin migrationsprocessmodell på. Detta för att dokumentationen i Amants och Stills rapport inte är så detaljerad på den punkten. Dock har det gått att se likheter mellan vad modellen tycker är viktigt att fokusera på och vad som har avrapporterats i de analyserade fallstudierna. Ett exempel på en likhet är att en analys är viktig att utföra för att inte implementera systemlösningar på ställen där det kan komma att uppstå problem. Ett annat exempel är att det är viktigt att upprätthålla en god dialog med de som är berörda av migrationen. Detta har varit viktigt i migrationsprojekten för att undvika irritation och motståndskraft från användarna. Det har inte gått att se några skillnader på om systemlösningmigrationen gällt en skola eller annan offentlig förvaltning då resultaten varit relativt lika. Det har heller inte spelat någon roll om systemlösningen gällt ett kontorspaket eller operativsystem. Detta för att det nya operativsystemsalternativen som varit aktuella upplevts som mycket användarvänliga.

Det går också att se likheter med de fördelar och nackdelar som Sers och Söderlund, (2009) kom fram till i sitt examensarbete. De skriver att risken för att betala licenskostnadsavgifter har undvikits genom att använda sig av det öppna källkodsalternativet OpenOffice istället för det dåvarande proprietära alternativet Microsoft Office. Samma sak har upplevts i de fallstudier som analyserats i den här studien. Det fanns dock skillnader i upplevda problem mellan Sers och Söderlunds (2009) resultat och de resultat som har presenterats i den här studien. Till exempel upplevde lärarna i Sers och Söderlunds (2009) studie att de hade kompatibilitetsproblem gällande att kunna återskapa dokument i både Microsoft Office och OpenOffice. Detta var inget problem som upplevdes i Bolzanoprovinsen eller hos det franska gendarmeriet. På grund av otillräcklig information i avrapporteringarna från Joinup (Joinup, 2102c) gällande den tekniska utformningen av projektens IT-miljöer går det inte se varför detta skiljde sig. Samma kompatibilitetsproblem upplevdes dock i migrationsprojektet i München. Det är möjligt att lärarna på skolan i Örebro kan dra lärdom av den lösning som applicerades i Münchens stad. Lösningen bestod i att sätta upp riktlinjer över hur ett dokument ska utformas för att det ska gå att återskapa på ett fullgott sätt i båda kontorspaketen.

Resultaten av analyserna visar att migrationsprocesserna har varit relativt enkla att genomföra, och projekten avrapporterar få större komplikationer som har uppstått under genomförandet. Detta är överraskande med tanke på projektens omfattning, organisationernas komplexitet samt det stora antalet datorer och användare som berördes.

Före migrationen betalade alla tre migrationsprojekten licensavgifter för proprietära systemlösningar. Det har nu visat sig att de aktuella förvaltningarna kan få tillgång till likvärdiga systemlösningar som är kostnadsfria att införskaffa. Detta är helt i enlighet med en av de ansvarspunkter som Statskontoret anser att en förvaltning bör följa, det vill säga att undvika onödiga utgifter. Migrationsprojektens rapporter visar alltså att det går att spara pengar vid migrationen. Detta är särskilt intressant eftersom fokus har legat på offentliga förvaltningar och det innebär då att det är skattepengar som sparas och kan läggas på andra viktiga saker inom den offentliga sektorn, till exempel vård, skola och omsorg.

Systemlösningar baserade på öppen källkod gör också att det blir lättare och säkrare att bevara viktiga dokument för framtiden. Förvaltningarna i migrationsprojekten har genom systemlösningmigrationerna rapporterat minskad risk för att filformat blir inaktuella och därmed också blir obrukbara. Dessa båda aspekter, den ekonomiska och den om bevaring av

dokument, är viktiga för samhället ur demokratisynpunkt. Medborgarna ska kunna vara säkra på att deras skattepengar används på bästa sätt och att offentliga dokument finns tillgängliga när någon efterfrågar dem. Den här studien har visat på att systemlösningar baserade på öppen källkod är positiva för samhället. Då avsikten är att IT-chefer i offentliga förvaltningar ska kunna använda sig av resultaten i den här studien är förhoppningen att detta leder till fler öppna källkodslösningar inom offentliga förvaltningar.

7 Slutsats

Resultatet av de utförda analyserna av de tre migrationsprojekten har visat att de risker som identifierats i projekten har upplevts på ungefär samma sätt. Detta kan bero på att de systemlösningar som varit aktuella för migrering har varit samma typ av systemlösning. Den risk som de tre projekten tycker skulle innebära störst konsekvens är risken att inte kunna ta sina egna IT-beslut, vilket innebär att förvaltningen mer eller mindre blir beroende av externa aktörer. Detta är en risk som de säger sig ha undvikit genom att de nu har systemlösningar som istället är baserade på öppen källkod. De risker som upplevts efter utförda systemlösningmigrationer kan sammanfattas med att det är risker som medför kortsiktiga problem för förvaltningarna. Det vill säga att förvaltningarna räknar med att riskerna kommer att minska och problem kommer att lösas allt eftersom tiden går. Exempelvis säger de tre migrationsprojekten att de har varit tvungna att anlita extern support för att lösa tekniska frågor relaterade till den nya systemlösningen. Dock säger de att de i framtiden räknar med att själva ansvara för supporten när personalens kompetens har höjts inom förvaltningarna.

De tre projektens migrationsprocesser har analyserats efter migrationsprocessmodellen skapad av Amant och Still (2007). Detta resulterade i att det gick att se en tydlig koppling mellan de tre projektens tillvägagångssätt vid utförd migration och vad migrationsprocessmodellen tycker är viktigt att fokusera på. Alla tre migrationsprojekten har rapporterat om en lyckad migrationsprocess och en tydlig koppling går att se mellan migrationsprocess och migrationsprocessmodell. Slutsatsen av detta är att en förvaltning som ska utföra en systemlösningmigration till öppen källkod kan minska antalet inträffade risker och problem om de följer de nio steg som *BRW migration model for desktop OSS* föreslår.

De redovisade migrationsprocesserna från varje migrationsprojekt kan också underlätta för en IT-chef som står i begrepp att genomföra ett liknande migrationsprojekt. Detta genom att det nu finns samlad information om lyckade migrationsprocesser i en och samma rapport.

Analyserna av de tre migrationsprojekten migrationsprocesser har också bidragit till att *BRW Migration Model for desktop OSS* har evaluerats ytterligare, vilket också efterfrågades av Amant och Still, skaparna av migrationsprocessmodellen (Amant och Still, 2007). Evalueringen av migrationsprocessmodellen bidrar även till en ökad kunskap gällande systemlösningmigrationer i allmänhet. Att studien är generellt utförd och inte fokuserar på enskilda systemlösningar gör att den också kompletterar de tidigare examensarbeten som har gjorts och som varit inriktade på mer specifika systemlösningar.

7.1 Framtida studier

För framtida studier skulle det vara intressant att se vilka risker som går att identifiera i migrationsprojekt med fokus på rättsliga frågor. Till exempel vad som gäller vid offentliga upphandlingar och hur detta skulle kunna påverka en migrationsprocess eller ett beslut inom ett migrationsprojekt.

Det skulle också vara intressant att se om det går att komplettera de identifierade riskerna som identifierats i de tre migrationsprojekten med risker från andra migrationsprojekt. En annan öppning för vidare studier kan då vara att analysera mindre migrationsprojekt än de som finns i den här studien. Detta för att se om upplevda risker och migrationsprocesser skiljer sig jämfört med de som finns presenterade i den här studien. Detta skulle innebära att

fler risker dokumenteras och medföra att dessa lättare kan undvikas i framtida migrationsprojekt.

För att ytterligare evaluera migrationsprocessmodellen som Amant och Still (2007) skapat, skulle det vara intressant att se om finns ytterligare steg som kan läggas till i modellen eller om det finns förslag till förbättringar. Att modellen i den här studien enbart testats på avrapporterade fallstudier gör också att det finns utrymme att testa modellen i samband med en systemlösningsmigration som utförs i praktiken, vilket skulle vara ett värdefullt komplement till den här studien.

7.2 Reflektion

När de utvalda migrationsprojekten analyserats och resultatet utformats har jag försökt att ha en så neutral ståndpunkt som möjligt. Min personliga åsikt är att jag tycker att systemlösningar baserad på öppen källkod ska användas i de fall där det är tekniskt möjligt, vilket med dagens kompetens och utveckling får anses vara fullt möjligt i de flesta IT-miljöer. På grund av att jag förespråkar ett användande av öppen källkod framför proprietära alternativ kan resultat och slutsats vara påverkade av detta. Ambitionen har dock varit att inte låta min personliga åsikt påverka resultatet av analysen och alla fakta finns väl refererade så att vem som helst kan kontrollera validiteten. Min personliga åsikt bör ändå beaktas när läsaren drar sina egna slutsatser av de presenterade resultaten.

De resultat som presenteras är alla baserade på den information som har gått att få fram i de tre migrationsprojekten. På grund av att det antagligen finns mycket mer information tillgänglig inom de båda migrationsprojekten på respektive förvaltning är det möjligt att resultaten sett annorlunda ut om informationen även funnits tillgänglig för nedladdning. Dock är sannolikheten stor att den då detaljerade informationen varit dokumenterad på tyska, franska och italienska, vilket hade försvårat analyserna avsevärt på grund av mina språkkunskaper. På grund av projektens storlek hade det antagligen också inneburit en alldeles för stor process att granska all information från tre så omfattande migrationsprojekt.

Som tidigare påtalats har de studerade migrationsprojekten genomförts med överraskande få komplikationer. Detta har påverkat min inställning positivt för att införa mer systemlösningar baserade på öppen källkod i offentliga förvaltningar. Jag är medveten om att det finns tekniska problem att lösa för att alla ska kunna övergå till liknande systemlösningar. Jag tror dock att problemen kommer att minska allt eftersom kunskapen ökar inom området.

Referenser

- Amant, K. S. och Still, B. (2007). *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. London: IGI Global. s.154-167.
- Berndtsson, M., Hansson, J., Olsson, B., och Lundell, B. (2008). *Thesis Projects: A Guide for Students in Computer Science and Information Systems*, 2nd ed, Springer Verlag.
- Cerri, D. och Fuggettas, A. (2007). Open standards, open formats, and open source. *Journal of Systems and Software*. 80 (11), s.1930–1937.
Tillgänglig på: www.sciencedirect.com/libraryproxy.his.se/science/article/pii/S0164121207000581 [Hämtad 21 februari 2012]
- Dalzien, J. (2003). Open Standards versus Open Source in E-learning, *Educause Quaterly*, 4, s.4-7. Tillgänglig på:
net.educause.edu/ir/library/pdf/eqm0340.pdf [Hämtad 21 februari 2012]
- Edwards, K. (2005). An economic perspective on software licenses—open source, maintainers and user-developers. *Telematics and Informatics*. 22 (1-2), s.111-133. Tillgänglig på: <http://www.sciencedirect.com/libraryproxy.his.se/science/article/pii/S0736585304000474> [Hämtad 21 februari 2012]
- Fuggetta, A. (2003). Open source software—an evaluation. *The Journal of Systems and Software*. 66 (2003), s.77-90. Tillgänglig på:
<http://www.sciencedirect.com/science/article/pii/S0164121202000651> [Hämtad 26 februari 2012]
- Hansen, M., Kohntopp, K. och Pfitzmann, A. (2002). The Open Source approach - opportunities and limitations with respect to security and privacy. *Computers & Security*. 21 (5), s.461–471. Tillgänglig på:
[http://dx.doi.org/libraryproxy.his.se/10.1016/S0167-4048\(02\)00516-3](http://dx.doi.org/libraryproxy.his.se/10.1016/S0167-4048(02)00516-3) [Hämtad 17 februari 2012]
- Hansson, M. och Larsson, M. (2009). Fria och öppna programvaror inom kommunal verksamhet – Vägen mot öppna standarder? Tillgänglig på: <http://hv.diva-portal.org/smash/get/diva2:226604/FULLTEXT01> [Hämtad 26 april 2012]
- Joinup (2012a). Declaration of Independence: The LiMux Project in Munich. *European Commission Joinup*. Tillgänglig på:
<https://joinup.ec.europa.eu/sites/default/files/studies/IDABC.OSOR.casestudy.LiMux.pdf> [Hämtad 22 februari 2012]
- Joinup (2012b). Towards the freedom of the operating system: The French Gendarmerie goes for Ubuntu. *European Commission Joinup*.
Tillgänglig på: <https://joinup.ec.europa.eu/sites/default/files/studies/IDABC.OSOR.casestudy.Gendarmerie.10.pdf> [Hämtad 23 februari 2012]
- Joinup (2012c). *European Commission Joinup*, Tillgänglig på:
<https://joinup.ec.europa.eu/>. [Hämtad 28 februari 2012]

- Joinup (2012d). Upgrade to Freedom - Schools in South Tyrol have their own GNU/Linux distribution. Tillgänglig på: <http://joinup.ec.europa.eu/elibrary/case/upgrade-freedom-schools-south-tyrol-have-their-own-gnu/linux-distribution-o> [Hämtad 1 juni 2012]
- Lakka, S., Michalakelis, C., Varoutas, D. och Martakos, D. (2012). Exploring the determinants of the OSS market potential: The case of the Apache web server. *Telecommunications Policy*. 36, s.51-68. Tillgänglig på: <http://www.sciencedirect.com.libraryproxy.his.se/science/article/pii/S0308596111002114> [Hämtad 23 april 2012]
- Merriam, S. B. (1994). Fallstudien som forskningsmetod. Lund: Studentlitteratur.
- OSI (1998a). Open Source Initiative. [Internet] Tillgänglig på: <http://www.opensource.org/docs/osd>. [Hämtad 21 februari 2012]
- OSI (1998b). Open Source Initiative. [Internet] Tillgänglig på: <http://www.opensource.org>. [Hämtad 21 februari 2012]
- Pfleeger, C. (2006). *Security in Computing*. Prentice Hall.
- Sers, A. och Söderlund, D. (2009). *Öppen mjukvarumigration. En studie om för- och nackdelar med en migration till öppen mjukvara*. Tillgänglig på: <http://oru.diva-portal.org/smash/get/diva2:318115/FULLTEXT01> [Hämtad 26 april 2012]
- Simon, K. D. (2005). The value of open standards and open-source software in government environments, *IBM Systems Journal*. 44(2), s.227-238. Tillgänglig på: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.7037&rep=rep1&type=pdf> [Hämtad 23 februari 2012]
- Statskontoret (2007). Den osynliga infrastrukturen – om förbättrad samordning av offentlig IT-standardisering, Statens Offentliga Utredningar: SOU 2007:47, Näringsdepartementet, Regeringskansliet. Tillgänglig på: <http://www.regeringen.se/content/1/c6/08/45/58/fd029160.pdf> [Hämtad 26 april 2012]
- Statskontoret (2003). Öppen programvara. Tillgänglig på: www.statskontoret.se/upload/publikationer/2003/200308.pdf [Hämtad 23 februari 2012]
- Winehq (2012). Winehq. [Internet] Tillgänglig på: <http://www.winehq.org/> [Hämtad 30 maj 2012]

Appendix A

Urval – migrationsprojekt tillgängliga på Joinup

Projektnamn:	Migreringen avser:	Berörda datorer:	Migrerade datorer:	Start:	Beräknat slut:
Franska gendarmeriet – Frankrike	Desktop	15 000	5 000	2008	2015
Münchens stad – Tyskland	Desktop	14 000	8 500	2005	2012
Finska justitiedepartementet – Finland	Desktop	10 000	8 500	2006	2007
Skolor i Bolzanoprovinsen – Italien	Desktop	2 460	2 460	2005	2007
Arbetsgivarnas arbetslöshetskassa – Tyskland	Desktop, Server	2 000	2 000	2003	2004
Kartläggnings- och fastighetsregistret – Tyskland	Desktop, Server	1 900	?	2005	2006
Beaumontsjukhuset – Irland	Desktop	1 000	?	2002	?
Katowices stad – Polen	Desktop	850	350	2005	?
Federala byrån för informationssäkerhet – Tyskland	Desktop	400	200	2002	2003
High school i Orwell – England	Desktop	300	300	2004	2005
Schwäbisch Halls stad – Tyskland	Desktop, Server	225	225	2001	2004
Schotens stad – Belgien	Desktop, Server	200	200	2005	2007
Prags kommun – Tjeckien	Desktop, Server	140	100	2001	2002
Fyra skolor – Norge	Desktop	100	100	2001	2003
Sjukhem i Rotterdam – Holland	Desktop	45	45	2003	2003
Amsterdams stad – Holland	Desktop, Server	35	35	2002	?
Artes stad – Frankrike	Desktop, Server	?	?	2004	2005
Bryssels huvudstadsregion – Belgien	Desktop	?	?	2004	?
Monopolkommissionen – Tyskland	Desktop	?	?	2002	2002
Fem kommuner i Groningen – Holland	Desktop, Server	?	?	?	?
Gdanskis stad – Polen	Desktop, Server	?	?	2001	2003
Federala domstolen – Schweiz	Desktop, Server	?	?	2001	2009
Salzburgs stad – Österrike	Desktop	?	?	2001	2009
Spanska ministeriet för offentlig förvaltning – Spanien	Server	1 250	?	2002	2004
Harens kommun – Tyskland	Server	225	225	2003	2005
Konkurrensmyndigheten – Tyskland	Server	140	?	2002	?
Rumäniens meteorologiska administration – Rumänien	Server	20	6	2002	2002
Svenska rikspolisstyrelsen – Sverige	Server	?	?	2005	2008
Norska kartinstitutet – Norge	Server	?	?	2006	2011
	Server	?	?	2007	?

Appendix B

Münchens stad

Det finns ett pågående projekt som startades år 2003 i Münchens stad i Tyskland som går under namnet LiMux. Det handlar om att alla offentligt anställda inom de administrativa avdelningarna i Münchens stad ska gå över från att använda proprietär programvara på sina arbetsdatorer till att använda sig av programvaror baserad på öppen källkod. Projektet omfattar cirka 14.000 arbetsdatorer. Anledningen till att de valde att starta detta projekt var att supporten för deras dåvarande operativsystem Microsoft Windows NT 4 skulle upphöra från Microsoft. De tvingades därför att se över sitt användande av operativsystem eftersom att det i slutändan skulle innebära att funktioner skulle sluta fungera (Joinup, 2012a). Münchens stad bad ett konsultföretag att se över vilka alternativ som fanns att tillgå. De ville se både lösningar med hjälp av Microsoft och lösningar baserad på öppen källkod. Resultatet av förstudien blev att det var ungefär lika i vilken lösning som var bäst lämpad.

Ett kostnadsförslag för en femårsperiod presenterades för varje lösning. Förslaget visade att det skulle kosta 35 miljoner euro med Microsofts förslag och 37 miljoner euro med öppna källkodslösningar. Båda förslagen inkluderade kostnader för upplärning och support. Om de skulle välja Microsofts lösning skulle det betyda att katalogtjänsten Active Directory skulle behöva användas vilket skulle resultera i en stor risk för inlåsning (Joinup, 2012). Även fast den proprietära lösningen skulle resultera i en något mindre kostnad beslöt de sig för att välja lösningen med den öppna källkoden. Främst var det för att de nu skulle kunna göra sina egna beslut kring IT-frågor utan att vara beroende av någon leverantör (Joinup, 2012a).

LiMux är även namnet på den GNU/Linux distribution som skulle användas på alla datorer och är baserad på Ubuntu (Joinup, 2012a). De programvaror som ytterligare skulle installeras på datorerna är:

Programvara	Förklaring
• K Desktop Environment (KDE)	Skrivbordsmiljö
• OpenOffice	Kontorspaket
• Firefox	Webbläsare
• Thunderbird	E-postklient
• Gimp	Bildbehandlingsprogram

Själva migreringsprocessen startade på respektive avdelning först när en utvärdering hade gjorts för att en migrering skulle vara genomförbar. Om en migrering inte skulle vara genomförbar på vissa programvaror skulle fortsatt användning av proprietära metoder användas. Dock föredrog Münchens stad en lösning på det problemet som bestod i att använda Windows som en virtuell dator på LiMux eller att använda det öppna källkodsemulatorn Wine som gör att ett Windows-program kan köras på LiMux (Joinup, 2012a). För att det inte skulle bli en allt för stor förändring på en gång fanns alternativet för avdelningarna att som första steg installera de öppna programvarorna på deras dåvarande Windowsdatorer. När de sedan bekantat sig med dessa kan de ta nästa steg till att byta ut operativsystemet till LiMux (Joinup, 2012a).

Appendix C

Franska Gendarmeriet

Det franska gendarmeriet är del av den franska polisen och arbetar med allt från vanlig brottsbekämpning till militära tjänster. Organisationen har mer än 100.000 anställda inom olika områden och är en av de största publika förvaltningarna i Europa. Tanken med migrering väcktes år 2001 då en grupp av IT-experter bestämde sig för att deras IT-system skulle behöva bli mer flexibelt för att kunna introducera nya standarder på ett lättare sätt (Joinup, 2012b). År 2005 togs ett beslut att byta ut deras nuvarande kontorspaket Microsoft Office till OpenOffice. Främsta anledningen till beslutet var att de inte ville att deras val av programvara skulle behöva tvinga dem till att använda ett specifikt operativsystem. Kort därefter, när OpenOffice hade installerats på de berörda datorerna, valde de också att introducera Mozillas webbläsare Firefox tillsammans med e-post klienten Thunderbird (Joinup, 2012b).

När Microsoft valde att introducera Windows Vista förstod Gendarmeriet att de skulle behöva byta sina nuvarande operativsystem inom en snar framtid. De beslöt då att de skulle byta ut skrivbordsdatorernas operativsystemplattform till en version av GNU/Linux. Deras servrar var sedan tidigare redan baserade på GNU/Linux och många av deras program var baserad på öppen källkod. Detta var en betydande del i deras kommande beslut att välja Ubuntu som distribution då det redan fanns personal som var kunniga inom området. Även det faktum att Ubuntu ansågs användarvänligt påverkade beslutet för val av distribution (Joinup, 2012b).

Efter en genomförd studie om vad det fanns för applikationer som skulle behöva ingå som standard på en skrivbordsdator kom de fram till en lista på 35 olika programvaror. De flesta programvaror gick att installera på Ubuntu, men de fann några kompatibilitetsproblem. Ett exempel som nämns är att det vid den tiden inte fanns ett fullgott alternativ för kalenderdelning som MS Outlook hade och som var mycket använt av några avdelningar. År 2008 har 5000 datorer migrerat till att använda Ubuntu som operativsystem och de räknade med att 15.000 datorer skulle vara migrerade år 2009. Gendarmeriet har räknat med att kunna spara 7 miljoner euro per år på att migrera till programvara baserat på öppen källkod.

Appendix D

Bolzanoprovinsen

Bolzano är en provins i norra Italien som har cirka 500.000 invånare. År 2004 visade det sig att de systemlösningar som fanns på datorerna inom Bolzanoprovinsens skolor behövde uppgraderas. Idén om att migrera de proprietära systemlösningarna till öppna källkodslösningar uppkom när en IT-lärare på en av skolorna pratade med en representant från skolmyndigheten i Bolzanoprovinsen och det visade sig att de båda delade samma vision om en öppnare IT-miljö. Skolmyndigheten tog då ett beslut att istället för att förnya systemlösningens licenserna så skulle de istället implementera öppna källkodsalternativ i skolorna. En annan anledning till migrationen var den funktionalitet som en systemlösning baserad på öppen källkod innebar. De ansåg att ett användande av öppen källkod skulle stärka elevernas kreativitet genom att de lättare skulle ha möjligheten att kunna utforska de öppna systemlösningarna än vad som var möjligt med de tidigare proprietära alternativen (Joinup, 2012d).

Migrationsprojektet startade år 2005 och gick under namnet FUSS (Free Upgrade South Tyrol's Schools). Projektet omfattade 72 skolor med tillhörande 2.460 skrivbordsdatorer och 80 servrar som alla skulle implementeras med systemlösningar baserade på öppen källkod. Som operativsystemplattform valdes Debian på grund dess rykte att vara ett stabilt och kvalitetssäkrat operativsystem. Migrationen utfördes under sommarlovet år 2005 då inga studenter fanns på skolorna. För att studenterna skulle få öva på att använda det nya operativsystemet skapades 20.000 LiveCDs som skickades hem till de berörda familjerna så att de alla kunde testa operativsystemet (Joinup, 2012d).

Kostnaderna för att utföra migrationsprojektet finansierades dels från den vanliga IT-budgeten som skolorna hade och dels från Europeiska socialfonden, som betalade för upplärningskostnaderna gällande lärare och tekniska supportteam. Kostnaden för migrationsprojektet beräknades till cirka 200.000 euro, vilket inte var högre än den tidigare årliga IT-kostnaden som låg runt samma nivå. Året därpå, år 2006, märkte de en stor skillnad på de ekonomiska utgifterna. De hade minskat drastiskt mot tidigare år. Till skillnad mot den tidigare kostnaden för proprietära systemlösningar på 200.000 euro, var nu summan 12.200 euro. Detta var pengar som gick till att betala anti-viruskydd till de kvarvarande Windows-datorerna och till den konsultfirma som hjälpt dem med tekniska problem. De säger också att de räknar med att utgifterna kommer att vara under 10.000 euro för nästkommande år, det vill säga år 2007. Att de valde att ha kvar ett antal datorer med Windows fortsatt installerat var för att det inte ansågs vara möjligt att migrerar de datorer som studenter med speciella behov nyttjade (Joinup, 2012d).