



LUNDS UNIVERSITET

Ekonomihögskolan

Open source på IT-företag

- FAKTORERNA SOM AVGÖR VALET ATT ANVÄNDA OPEN SOURCE ELLER INTE

Kandidatuppsats, 15 högskolepoäng, SYSK02 i informatik

Framlagd *Maj 2014*

Författare: *Fredrik Källström*
Björn Svensson

Handledare: *Anders Svensson*
Examinatorer: *Odd Steen*
Mirella Muhic

Titel:	Open source på IT-företag – Faktorerna som avgör valet att använda open source eller inte.
Författare:	Fredrik Källström Björn Svensson
Utgivare:	Institutionen för informatik
Handledare:	Anders Svensson
Examinatorer:	Odd Steen Mirella Muhic
Publiceringsår:	2014
Uppsattstyp:	Kandidatuppsats
Språk:	Svenska
Nyckelord	Open source, closed source, mjukvara

Abstrakt

Open source är idag ett vida känt begrepp och det finns numera ett open source alternativ till nästan varje mjukvara av proprietär sort. Med en växande användarbas och en kostnad som nästan alltid är obefintlig är det naturligt att företag ser open source som ett alternativ till att köpa proprietär mjukvara. Genom denna uppsats undersöker vi därför vilka egenskaper hos open source som är viktiga och mindre viktiga när beslut om att använda den typen av mjukvara tas, och om IT-företag idag över huvud taget överväger att använda open source. För att svara på detta skapas ett teoretiskt ramverk med totalt sex faktorer som baserat på tidigare forskning och litteratur är de vanligast förekommande egenskaperna hos open source. Fyra stycken intervjuer genomförs sedan med små till medelstora IT-företag där frågorna baseras på ramverkets sex faktorer.

Innehållsförteckning

Innehållsförteckning.....	3
Figurförteckning.....	4
Tabellförteckning.....	4
1. Introduktion	5
1.1 Problemdefinition & Forskningsfråga	6
1.2 Syfte	7
1.3 Avgränsning	7
1.4 Förkortningar & uttryck	8
2. Litteraturgenomgång & Teoretiskt ramverk.....	9
2.1 Teoretiskt ramverk.....	9
2.1.1 Definition av faktorer i ramverket.....	11
2.3 Litteraturstudie av faktorer	13
2.3.1 Grad av buggar och sårbarheter	13
2.3.2 Licenskostnad & supportkostnad	16
2.3.3 Framtidssäker investering	17
2.3.4 Öppen källkod.....	18
2.3.5 Möjlighet att skraddarsy mjukvara	19
2.3.6 Bidrag av och till community	19
3. Metod.....	21
3.1 Datainsamling.....	21
3.1.1 Urval och informanter	21
3.1.2 Intervjuguide.....	22
3.2 Analys.....	23
3.3 Etik	23
3.4 Kritik av metodval	23
4. Empiri.....	24
4.1 Presentation av intervjuer	25
4.2 Användning och uppfattning av OSS	25
4.3 Svar efter kategori i ramverk.....	27
4.3.1 Säkerhet.....	27
4.3.2 Ekonomi	29
4.3.3 Utveckling.....	30
5. Diskussion	31
6. Slutsats.....	35
7. Referenser	37
8. Bilagor	40
8.1 Bilaga 1 - Intervjumall	40
8.2 Bilaga 2 - Intervju Fishbone	42
8.3 Bilaga 3 – Intervju Force 12	47
8.4 Bilaga 4 – Intervju Lunicore	53
8.5 Bilaga 5 – Intervju Företag 4.....	59

Figurförteckning

Figur 1 - Illustration av ramverkets uppbyggnad	9
Figur 2 - Teoretiskt ramverk.....	10
Figur 3 - Marknadsandelar operativsystem (Netmarketshare, 2014).....	17
Figur 4 - Antal bidrag per utvecklare (Lerner & Tirole, 2002).....	20

Tabellförteckning

Tabell 1 - Sårbarheter per mjukvara (Schryen & Rich, 2010)	13
Tabell 2 - Utvecklingstyp per mjukvara. Modifierad från Schryen & Rich (2010).....	14
Tabell 3 - Buggar per 1000 rader kod (Mockus et al., 2000)	15
Tabell 4 - Detaljer över intervjudeltagare	22
Tabell 5 - Deltagande informanter och företag	24
Tabell 6 - Relation mellan faktorer i intervju och teoretiskt ramverk.....	24
Tabell 7 - Rangordnade faktorer av informanter	25
Tabell 8 - Rangordnade faktorer i säkerhetskategorin.....	27
Tabell 9 - Rangordnade faktorer i ekonomikategorin.....	29
Tabell 10 - Rangordnade faktorer i utvecklingskategorin	30

1. Introduktion

Kapitel ett ger en kort bakgrund till ämnet open source. Problemet uppsatsen grundas på presenteras, syfte och avgränsningar fastställs och slutligen listas och förklaras förkortningar till ofta förekommande termer i uppsatsen.

För 15-20 år sedan var open source software (OSS) ett begrepp som inte var allmänt känt. Företag som funderade över att införa ny mjukvara i sin organisation vände sig ofta till proprietära alternativ. Denna uppsats belyser först OSS intåg på marknaden samt dess framfart mot att idag vara en stark utmanare till proprietära alternativ. I denna uppsats ämnar vi att i huvudsak undersöka vilka faktorer som är viktigast för IT-företag när de väljer en OSS framför en proprietär mjukvara för att användas i verksamheten.

Tankesättet som OSS grundar sig i härstammar från 1960 - 1970-talet, då både forskare och ingenjörer inom företag och den akademiska världen modifierade och utbytte mjukvara med varandra (Carillo & Okoli, 2008). Det var således inte med Linux födelse som den fundamentala ideologin bakom OSS uppstod. Användandet av OSS har dock kommit en lång väg sedan Linus Torvalds skapade operativsystemet Linux, byggt med öppen källkod under 1990-talets början.

Under de senare åren på 1990-talet fick Linux sitt erkännande bland allmänheten genom publikationer i bland annat Forbes, samtidigt släpptes källkoden fri för webbläsaren Netscape (Open Source Initiative, 2012) (McHugh, 1998). En kort tid efter dessa händelser myntades benämningen "open source software" genom att organisationen Open Source Initiative (OSI) grundades i Palo Alto, Kalifornien år 1998 (Open Source Initiative, 2012). Beslutet att släppa källkoden för Netscape fri influerades till stor del av Eric S. Raymonds essä "The Cathedral and the bazaar" från 1997 (Open Source Initiative, 2012). Raymond hade länge velat se att världen fick upp ögonen för open source, därför var han en drivande kraft bakom bildandet och till slut även medgrundare av OSI, han blev tillika organisationens första President (Open Source Initiative, 2012).

Som en reaktion till Netscapes tanke att släppa sin källkod fri uttalade sig Raymond i en intervju att: "*Netscape doing this creates a window of opportunity for us to get our message into corporate boardrooms*" (Leonard, 1998). Dessa samlade händelser hade nu gett Raymond och OSI ett fäste, det var nu på allvar möjligt att utmana proprietär mjukvara och få in foten i företagsvärlden.

I april 1995 släppte The Apache Group den första publika versionen av Apache Web-Server, byggd med öppen källkod. Apache Web-Server byggdes på grunden från den redan befintliga webbservern NCSA httpd 1.3. Inom ett år hade Apache passerat NCSA som den mest använda web-servern (Apache Software Foundation, 2014).

Enligt Netcraft (2014) befann sig Apache under år 2005 i sina glansdagar, då uppnådde företaget häpnadsväckande 71 % marknadsandel utav driften av världens webbsidor. I den senast utförda undersökningen visar resultatet att Apache gjort en kraftig förlust i marknadsandelar till fördel för Microsoft. I april 2014 var gapet mellan de två

konkurrenterna liten, 38 % marknadsandel för Apache respektive 33 % för Microsoft. Övriga marknadsandelar är fördelade på; nginx 15 %, Google 2 %, samt en ospecificerad mängd övriga aktörer 12 %.

Redan år 1999 såg IBM potential för framgång i Linux, genom att gå in med finansiering och stöd kunde chanserna för framgång öka. I Linux såg IBM chansen att få ett enhetligt operativsystem på sina plattformar, deras tankesätt var att få Linux mer anpassat mot företagsmiljöer (Samuelson, 2006). Samtidigt kunde IBM undvika att utveckla ett eget operativsystem från grunden, och med användandet av Linux tilläts IBM att vara oberoende av Microsofts licensvillkor (Capek, Frank, Gerdt & Shields, 2005).

Denna anpassning mot företagsmiljöer gjordes bland annat genom att IBM donerade några av komponenterna från sin proprietära mjukvara, AIX, till Linux (Samuelson, 2006). Sedan år 2001 har IBM finansierat utvecklingen av Linux med mer än 1 miljard USD, samtidigt har även andra open source projekt fått stöd och finansiering (Samuelson, 2006; IBM, 2011 & 2013; Capek et al., 2005).

Idag har OSS ett starkt fotfäste på mjukvarumarknaden. I en undersökning utförd av Gartner (2011) deltog 547 IT-chefer från organisationer i elva länder. Undersökningen visade att 22 % av de tillfrågade organisationerna använde OSS i hela sin verksamhet, medan 46 % av organisationerna använde OSS inom specifika avdelningar och projekt (Gartner, 2011). I november år 2010 drevs 459 utav världens 500 högst rankade superdatorer av en version av Linux operativsystem (IBM, 2011).

Begreppet OSS är inte längre ett okänt fenomen, det är snarare en stark utmanare till proprietära alternativ. Företag idag kan således få en konkurrenskraftig valmöjlighet när det kommer till införandet av ny mjukvara i sin organisation. Denna valmöjlighet innebär att ett beslut måste utgå från vissa faktorer och ett resonemang.

1.1 Problemdefinition & Forskningsfråga

Sedan Raymonds essä "The cathedral and the bazaar" publicerades 1997, senare även publicerad som bok (Raymond, 2001) har något hänt på mjukvarumarknaden. Den skapade en revolution inom open source världen och allt fler börjande anamma den öppna utvecklingsstrategin som Raymond definierat. Open source har sedan dess blivit populärare, det har vuxit från att bara användas på persondatorer till att bli den dominerande produkten på flera områden, Apache bland webbserver etc. Det får idag helt enkelt ses som ett reellt alternativ för företag att använda. Inte bara för att det generellt är gratis, utan även för att mjukvarorna ofta är minst lika bra. Men med alternativ försvåras beslutsfattandet, nu när den proprietära marknaden utmanas av open source marknaden finns det två alternativ av mjukvarustrukturer. För företag krävs det att de är medvetna om styrkor och svagheter hos de olika alternativen.

Dessa styrkor och svagheter kan vara svåra att reda ut. De traditionella proprietära alternativen har alltid varit ett säkert kort, men är de verkligen bättre. Eller kan en mjukvara som är gratis även vara bättre? är inte det gamla uttrycket "du får vad du betalar för" giltigt även på mjukvaror?

Det är vad vi i denna uppsats ämnar undersöka, vilka faktorer kopplade till OSS är viktigast för IT-företag när de tar beslut om att använda en OSS och inte en proprietär motsvarighet.

Forskningsfrågorna uppsatsen ämnar svara på blir därför:

*Vilka faktorer är avgörande när IT-företag väljer en open source-mjukvara framför de proprietära alternativen?
- Är open source över huvud taget ett reellt alternativ för dessa företag?*

Dessa kommer att besvaras genom att ett teoretiskt ramverk bestående av faktorer tas fram grundat på tidigare litteratur och forskning. Detta ramverk sätts sedan på prov när vi genomför ett antal intervjuer med företag inom IT-branschen där frågor kopplat till ramverket ställs, svaren förankras sedan i ramverkets faktorer.

1.2 Syfte

Denna uppsats avser att genom en empirisk studie undersöka hur IT-företag resonerar kring valet att använda open source i sin arbetsmiljö. Vilka faktorer detta beslut grundas på samt om open source-mjukvaror ens finns i åtanke när ett beslut ska tas. Den empiriska studien förankras i ett teoretiskt ramverk grundat på tidigare litteratur och forskning på ämnet.

1.3 Avgränsning

Undersökningen är avgränsad till mjukvara som bestämts av företaget och dess avdelningar och inte av individen själv för personligt bruk. En anställds val att exempelvis använda en specifik open source-webbläsare är således inte av intresse för denna uppsats.

Open source kan av vissa vara knutet till en kulturell aspekt, där användandet av open source är starkt kopplat till transparensen i produkten och frihet från stora kommersiella företag, vilket ses som något negativt. Denna kulturella aspekt bortser denna uppsats helt ifrån.

Undersökningen kommer inte ta företagens och individens kunskap om open source i beaktning. Det krävs dock att en generell förståelse finns för vad termen open source innebär.

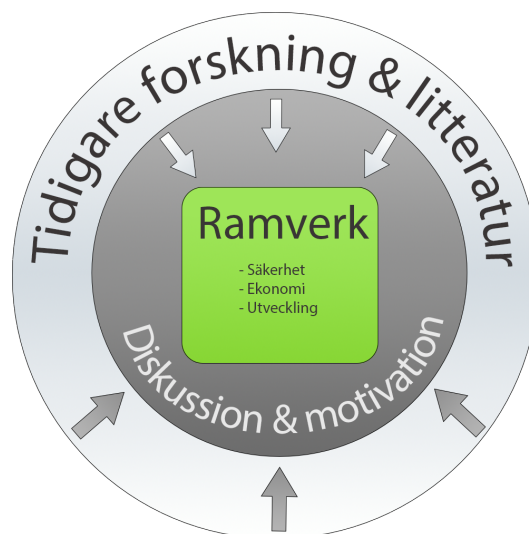
1.4 Förkortningar & uttryck

- **Alpha** (Mjukvaruversion):
Tidig version av en mjukvara, ofta ostabil och otestad.
- **Beta** (Mjukvaruversion):
Sista versionen av en mjukvara innan den blir släppt publikt. Är oftast testad i miljöer som skall efterlikna verkligheten.
- **Closed Source Software: CSS.**
Proprietär mjukvara. Refererar till all mjukvara som innefattar restriktioner gällande användning, editering, kopiering eller distribuering av modifierade versioner.
- **Community:**
Community hänvisar till den grupp utvecklare som bidrar och använder sig av en viss open source-mjukvara.
- **Freeware:**
Mjukvara som är gratis att använda, samt som får distribueras och spridas fritt. Denna typ av mjukvara saknar öppen källkod och är skyddad av upphovsrätt.
- **GNU General Public License: GPL**
En utav de mest använda licenserna för open source-mjukvara.
- **Källkod:**
Instruktioner och information skriven i något programmeringsspråk. Koden kompileras och exekveras för att bli ett körbart program för datorn.
- **Open Source Initiative: OSI.**
Open Source Initiative är ett icke-vinstdrivande bolag med global räckvidd bildat för att utbilda om och förespråka för fördelarna med öppen källkod samt att bygga broar mellan olika delar inom open source-rörelsen.
- **Open Source Software: OSS.**
Mjukvara byggd med öppen källkod, som får distribueras utan begränsningar, samt där det är fritt fram att editera källkoden och distribuera nya versioner utav den.
- **Patch, patcher:**
Patch innebär en uppdatering till en mjukvara i syfte att lösa ett specifikt problem/bugg.
- **Total Cost of Ownership: TCO**
En metod för att analysera samtliga kostnader knutna till en IT-investering.

2. Litteraturgenomgång & Teoretiskt ramverk

Kapitel två presenterar det teoretiska ramverket samt den litteratur och tidigare forskning som styrker ramverkets faktorer. I delkapitel 2.1 motiveras först varför ramverket har upprättats, sedan illustreras ramverket i figur 2. I delkapitel 2.1.1 förklaras och definieras ramverkets sex faktorer. Dessa sex faktorer återkommer sedan i kapitel 2.3 där litteratur och tidigare forskning presenteras enskilt för varje faktor i ramverket.

2.1 Teoretiskt ramverk



Figur 1 - Illustration av ramverkets uppbyggnad

Sedan Raymond (2001) publicerade den för den tiden nyskapande boken "The Cathedral and the Bazaar" har ämnet open source undersökts flitigt i mängder med olika syften, och inte sällan i relation till proprietär mjukvara. Bland tidigare undersökningar, teorier och publikationer inom ämnet fann vi dock inga där hela spannet av faktorer behandlas. Samtliga behandlade någon specifik eller några enstaka delar av fenomenet open source, där det ofta försöks förklaras i relation till proprietär mjukvara. Vi valde därför att definiera ett eget ramverk med tre kategorier som baserats på de karaktärsdrag hos open source som var vanligast förekommande i tidigare publicerat material på området. Det resulterade i kategorierna Säkerhet, Ekonomi och Utveckling. Kopplade till dessa kategorier fann vi totalt sex faktorer, det tidigare publicerade materialet kring dem som används i denna uppsats presenteras här kortfattat för att sedan i kapitel 2.3 utforskas mer utförligt.

Den fria källkoden och de fördelar och nackdelar det skapar är kanske den tydligaste skillnaden mellan open source och closed source, och den faktor (Öppen källkod, kap. 2.3.4) som belysts mest i tidigare litteratur. Dedrick & West (2006) undersökte bland annat genom att intervjua 21 personer med inblick i totalt 14 organisationer huruvida tillgång till källkod är av intresse eller ej. Pizka (2004) däremot genomför ett exempel där två likvärdiga projekt försöks genomföras, ett där en mjukvara skapas från grunden

och ett där den fria källkoden från en OSS utnyttjas som en plattform för vidareutveckling, för att se vilket tillvägagångssätt som är effektivast sett till utvecklingstid.

Lerner & Tirole (2002) undersökte exempelvis varför utvecklare väljer att arbeta med open source utan någon direkt betalning i gengäld, även Krishnamurthy (2002) undersökte karaktären av bidragare till open source-projekt.

Säkerheten hos open source och closed source har även undersökts flitigt, av Schryen & Rich (2010) bland andra, genom en empirisk undersökning som jämför antalet rapporterade sårbarheter hos 17 olika OSS och CSS inom olika områden. Även Mockus, Fielding & Herbsleb (2000) gjorde en liknande undersökning på det välkända open source-projektet Apache och fyra ej namngivna closed source-projekt där de analyserade antalet buggar per 1000 rader kod.

Den ekonomiska faktorn är naturligtvis intressant när det kommer till open source och closed source, det är något som bland annat Dedrick & West (2006) kunde konstatera efter deras intervjuer. Daffara & Gonzalez-Barahona (2000) behandlar de skillnader i de ekonomiska modellerna som open source skapar mot de traditionella proprietära motsvarigheterna. De tar framförallt upp och jämför begreppet "Total Cost of Ownership" vilket ser till den totala kostnaden och inte bara inköpskostnaden av en mjukvara.

All denna tidigare litteratur, forskning, undersökningar och teorier tyder på att det är de sex faktorerna som presenteras i figur 2 nedan över vårt ramverk som är de viktigaste hos open source. De sex faktorerna delade vi in i de tre kategorierna Säkerhet, Ekonomi och Utveckling. I kapitel 2.3 ges en utförlig genomgång av tidigare litteratur och forskning kopplat till varje faktor, vilket kapitel varje faktor tillhör visas inom parenteser i figuren nedan.



Figur 2 - Teoretiskt ramverk

2.1.1 Definition av faktorer i ramverket

Varje faktor i ramverket ges här en kort introduktion, där egenskaperna hos varje faktor tas upp. I kapitel 2.3 ges faktorerna sedan en utförlig genomgång baserad på tidigare forskning, inom parenteser visas det till vilket delkapitel faktorn hör i kommande kapitel.

Grad av buggar och sårbarheter (2.3.1)

Möjligheten att granska och editera källkoden i OSS innebär att det lämnas möjligheter för vem som helst att finna problemområden, säkerhetshål och brister i källkoden och även att korrigera dessa. I kontrast till att brister och säkerhetshål lämnas synliga för att korrigeras så är de lika synliga för intressenter som inte vill väl (Schryen & Rich, 2010). Någon med syfte att attackera mjukvaran kan identifiera svagheter i koden på samma sätt och använda de svagheter för att attackera programmet och de som använder det. Genom att granska antalet buggar som utgör en sårbarhet och dess lösningsgrad i en mjukvara kan nivån av säkerhet fastställas i hänseende till säkerhet i oförutsedda säkerhetsbrister (Schryen & Rich, 2010).

När vi resonerar kring en mjukvaras kvalitét i denna uppsats, syftar vi på andelen defekter i dess kod. Detta menar vi vara ett tydligt mått för att kunna ställa olika mjukvarualternativ i relation till varandra gällande deras kvalitét.

Licenskostnad & supportkostnad (2.3.2)

En utgångspunkt i jämförelsen mellan OSS och CSS är kostnaden för licenser. Daffara & Gonzalez-Barahona (2000) påstår att ett företag med mindre ekonomiska medel och begränsat eget kapital för framtida investeringar kan skapa sig en fördel med hjälp av OSS. Kostnaden för att införskaffa OSS är visserligen obefintlig men däremot kan det tillkomma utvecklingskostnader för att analysera kvalitén och säkerheten samt tillhandahålla support efter det att communityn inte längre är aktiv.

Framtidssäker investering (2.3.3)

Strukturen för support av OSS skiljer sig kraftigt från vanlig proprietär mjukvara där ett avtal ingås med leverantören som ofta specificerar exakt vilken typ av support som lämnas, hur länge produkten (mjukvaran) täcks av deras support och det är även praxis att leverantören uppdaterar mjukvaran kontinuerligt och vid behov. I OSS däremot finns generellt inga supportavtal, licenser och framförallt inga garantier. Utveckling av denna typ av mjukvara görs frivilligt av communityn och kan när som helst upphöra. Enligt Lerner & Tirole (2002) möjliggör dock den öppna källkoden i OSS att fortsatt support kan ges efter livslängdens slut, något som Tripathi & Tiwari (2013) hävdar inte är fallet när det gäller proprietär mjukvara. En proprietär mjukvara, såsom Microsoft Windows, har således en fast livslängd medan en OSS som exempelvis Linux lever på obestämd tid.

Öppen källkod (2.3.4)

Den öppna källkoden i OSS innebär att vem som helst får tillgång till programkoden, den kod som utgör själva programmet och dess funktioner. Den öppna källkoden är en förutsättning för att en mjukvara skall klassificeras som OSS (Open Source Initiative, 2014b). Den öppna källkoden möjliggör för editering av mjukvaran och gör mjukvaran mycket dynamisk, mjukvaran kan teoretiskt granskas i minsta detalj för att finna absolut alla brister och buggar för att sedan även tillåta möjligheten att korrigera dem. Om mjukvaran saknar en funktionalitet är det möjligt att utveckla denna själv och editera originalversionen för att inkludera den nya funktionen. (Fitzgerald, 2006)

Raymond (2001) nämner en annan styrka med tillgång till källkod, möjligheten för användarna att försäkra sig om att mjukvaran inte utför något utöver dess utlovade funktionalitet, och att den utför funktionerna på ett säkert sätt. I den traditionella proprietära mjukvarustrukturen där källkoden hålls hemlig får användaren bara tillgång till ett exekverat och körbart program. Detta innebär att inga andra än utvecklarna kan se hur programmet fungerar egentligen, användarna kan bara se vilka funktioner programmet har i användargränssnittet. Teoretiskt sett kan det alltså döljas funktioner som utförs i bakgrunden och som hålls hemlig för användarna.

Möjlighet att skräddarsy mjukvara (2.3.5)

Med proprietär mjukvara får användaren acceptera produkten för vad den är och vad den kan utföra, om den inte möter alla behov kan inte mjukvaran i sig editeras. Ibland kan en sådan situation lösas genom ytterligare mjukvaror, egenutvecklade eller inte, som kopplas samman för att tillföra funktionalitet. OSS kan däremot utvecklas och byggas vidare utöver dess ursprungliga funktionalitet. Det krävs inte ytterligare programvara eller tillägg, ny funktionalitet kan adderas direkt i källkoden. (Fitzgerald, 2006).

Bidrag av och till community (2.3.6)

En stark fördel med OSS är möjligheten det skapar för dess community. Mjukvaran kan utvecklas av en bred mängd utvecklare. Det finns alltså en möjlighet att många utvecklare granskar koden, och när den blir granskad av många bör bristerna rimligtvis också upptäckas och åtgärdas fortare, som uttryckt av Raymond "*given enough eyeballs, all bugs are shallow*" (2001, s. 30). Det finns också en faktor av eget bidrag till mjukvaran. När nya funktioner eller buggfixar behövs i vanlig proprietär mjukvara kan endast en önskan lämnas till leverantören. Används OSS kan istället företaget själv utveckla de nya funktionerna eller korrigera de buggar de finner, om de sedan offentliggör sin utvecklade programvara drivs OSS programvaran framåt och de hjälper communityn i stort.

2.2 Litteraturstudie av faktorer

I följande delkapitel utforskas varje faktor utförligt var för sig baserat på tidigare publicerat forskningsmaterial och teorier.

2.2.1 Grad av buggar och sårbarheter

Att generellt påstå att OSS är säkrare med färre fel eller vice versa vore felaktigt, detta är ett påstående som endast kan göras efter en granskning av mjukvaran i detalj. Genom att granska antalet buggar, och framförallt buggar som utgör en sårbarhet i en mjukvara, ges ett underlag för att se nivån av säkerhet hos en mjukvara, åtminstone i hänseendet om säkerhet i oförutsedda säkerhetsbrister. (Schryen & Rich, 2010).

Sådan granskning har gjorts av bland annat Schryen & Rich (2010) som undersöker 17 olika mjukvaror, varav åtta är open source och nio är proprietära. Undersökningen visar på antalet funna sårbarheter i de olika mjukvarorna samt antalet sårbarheter som blivit lösta av patcher. Endast sårbarheter som är accepterade av ramverket Common Vulnerabilities and Exposures (CVE) som upprättats av Mitre (2014) inkluderas i undersökningen. En bugg i en av mjukvarorna inkluderas därför inte om den inte är definierad som en sårbarhet i CVE av Mitre. Även allvarlighetsgraden definieras per sårbarhet i en skala från ett till tio där ett är minst allvarlig och tio är mest allvarlig (Schryen & Rich, 2010).

Tabell 1 - Sårbarheter per mjukvara (Schryen & Rich, 2010)

Application type	Product	Vulnerabilities (un)patched			Median of severities		
		#vuln.	#vuln. unpatched	Prop. of unpatched vuln.	unpatched	patched	overall
Browser	Internet Explorer 7	74	49	66.22%	5.0	9.3	6.8
	Firefox 2	167	34	20.36%	5.0	6.8	6.4
Email client	MS Outlook Express 6	23	15	65.22%	5.0	7.3	5.1
	Thunderbird 1	110	6	5.45%	3.45	6.95	6.8
Web server	IIS 5	83	40	48.19%	5.0	7.2	5.0
	Apache2	80	21	26.25%	4.7	5.0	5.0
Office	MS Office 2003	99	4	4.04%	5.05	9.3	9.3
	OpenOffice 2	19	4	21.05%	5.25	9.3	7.6
Operating system	Windows 2000	385	117	30.39%	5.1	7.2	7.2
	Windows XP	297	91	30.64%	5.0	7.5	7.2
	MAC OSX	300	20	6.67%	5.0	6.8	6.8
	Red Hat Enterprise Linux 4	264	39	14.77%	4.9	4.9	4.9
	Debian 3.1	207	30	14.49%	4.9	4.9	4.9
Database Management System	MySQL 5	33	8	24.24%	4.6	4.9	4.9
	PostgreSQL 8	25	3	12.00%	9.0	6.3	6.8
	Oracle 10g	63	8	12.70%	7.35	5.5	5.5
	DB2 v8	13	1	7.69%	7.8	7.2	7.2

Tabellförklaring: Tabell 1 visar antal funna sårbarheter(#vuln) och hur många av dem som har lösts(#vuln unpatched). Vilket resulterar i lösningsgraden (Prop. of unpatched vuln.)

Tabell 2 - Utvecklingstyp per mjukvara. Modifierad från Schryen & Rich (2010)

Application type	Product	Devel. Type ¹⁾
Browser	Internet Explorer 7	Closed
	Firefox 2	Open (BS)
Email client	MS Outlook Express 6	Closed
	Thunderbird 1	Open (CS)
Web server	IIS 5	Closed
	Apache2	Open (CS)
Office	MS Office 2003	Closed
	OpenOffice 2	Open (CS)
Operating system	Windows 2000	Closed
	Windows XP	Closed
	MAC OSX	Closed ⁴⁾
	Red Hat Enterprise Linux 4 ³⁾	Open (CS)
	Debian 3.1 ³⁾	Open (BS)
Database Management Systems	mySQL 5	Open (BS)
	PostgreSQL 8	Open (CS)
	Oracle 10g	Closed
	DB2 v8	Closed

Schryen & Rich (2010) visar att det finns tydliga skillnader inom vissa applikationstyper, bland operativsystem exempelvis visar de två OSS på en bättre proportion av opatchade sårbarheter, åtminstone i jämförelse mot de två Microsoft produkterna Windows XP & 2000. Inom applikationstyperna webbläsare och email-klienter visar tabellen att de två OSS projekten Firefox och Thunderbird har en mycket högre grad av rapporterade sårbarheter än deras CSS motsvarigheter. Däremot har de även en mycket högre grad av patchade sårbarheter, och proportionen av sårbarheter som inte blivit patchade är till OSS projektens fördel.

Överlag visar dock tabellen på jämna nivåer både på antalet rapporterade sårbarheter som proportionen av patchade sårbarheter. Slutligen visar tabellen även att inom i stort sett alla mjukvaror, förutom Debian och Red Hat Enterprise Linux, patchas främst de sårbarheterna med hög allvarlighetsgrad och de sårbarheterna med låg allvarlighetsgrad lämnas opatchade (Schryen & Rich, 2010).

Även Mockus et al. (2000) undersöker OSS i relation till CSS, deras undersökning visar antalet fel per tusen rader kod. Detta görs i totalt fyra mätningar inom två kategorier. Dels en kategori där endast fel upptäckta efter att mjukvaran släppts publikt inkluderas (Post-release) och den andra kategorin där även fel under testfasen av mjukvaran inkluderas. Båda kategorierna mäts på två sätt, KLOCA vilket mäter fel per tusen rader tillagd kod. Och KDelta vilket är de kvarvarande raderna kod efter tillagda rader och borttagna rader.

Tabell 3 - Buggar per 1000 rader kod (Mockus et al., 2000)

Measure	Apache	A	C	D	E
Post-release Defects/KLOCA	2.64	0.11	0.1	0.7	0.1
Post-release Defects/KDelta	40.8	4.3	14	28	10
Post-feature test Defects/KLOCA	2.64	*	5.7	6.0	6.9
Post-feature test Defects/KDelta	40.8	*	164	196	256

Tabellbeskrivning: Tabellen visar buggar per tusen rader kod i KLOCA och KDelta för en open source-mjukvara och fyra anonymiserade closed source-mjukvaror (A, C, D & E). KLOCA = fel per tusen rader tillagd kod. KDelta = fel per tusen rader kod av delta (tillagda rader kod - borttagna rader kod)

Genom tabell 3 visar Mockus et al. (2000) på tydliga skillnader i rapporterade fel per tusen rader kod på både KDelta och KLOCA i Post-release där Apache mjukvaran har fler rapporterade fel. I den andra kategorin där även fel under testfasen inkluderas svänger resultatet till Apache mjukvarans fördel och där har istället de proprietära mjukvarorna fler rapporterade fel i både KDelta och KLOCA.

Undersökningen gjord av Schryen & Rich (2010) visar på små skillnader hos OSS och CSS. Undersökningen av Mockus et al. (2000) visar däremot på relativt stora skillnader. Dock inkluderar den senare endast en OSS och det kan därför inte anses vara ett representativt resultat för OSS överlag. Något de båda undersökningarna gemensamt visar på är att de OSS som undersökts har haft en högre lösningsgrad av buggar och sårbarheter. Det är faktiskt så högt att sju av nio inkluderade OSS i de båda undersökningarna har en bättre lösningsgrad än deras proprietära motsvarigheter (Schryen & Rich, 2010) (Mockus et al., 2000).

Organisationen Coverity gör årligen en stor undersökning på ett stort antal mjukvaror och rapporterade år 2013 att dem OSS de undersökte hade ett genomsnitt om 0.59 defekter per 1000 rader kod. Det ställer dem gentemot 0.72 defekter hos de proprietära mjukvaror de undersökte (år 2012 rapporterades OSS ha 0.59 och CSS ha 0.58). Enligt rapporten har alltså en stor aspekt av kvalitén hos OSS projekt ökat de senaste åren (Coverity, 2014). Något som styrker detta ytterligare är även webbsidan SourceForge där en stor del av alla open source-projekt publiceras, de ger mjukvarorna en status baserad på deras mognadsgrad. Mjukvarorna kan få status "Planning", "Pre-alpha", "Alpha", "Beta", "Production/stable", "Mature" eller "Inactive", där "Production/stable" innebär att det är en version som är släppt publikt och "Mature" att det är en version som har funnits som publik release ett tag. I maj 2002 hade SourceForge 4365 projekt med status "Production/Stable" och 480 projekt med status "Mature" av totalt 29 274 st. projekt (Krishnamurthy, 2002). I maj 2014 har samma webbsida 40 850 projekt med status "Production/Stable" och 3644 projekt med status "Mature" av totalt 188 671 st. projekt (Sourceforge, 2014a). Andelen projekt med status "Production/Stable" har ökat med $\approx 6,8\%$ och andelen i status "Mature" med $\approx 0,3\%$. Den totala mängden open source-projekt har ökat markant mellan dessa år, men siffrorna visar ändå på att andelen kvalitativa OSS är högre år 2014 än vad den var år 2002.

2.2.2 Licenskostnad & supportkostnad

Mjukvara som klassificeras som OSS behöver inte nödvändigtvis distribueras gratis. En av de mer populära OSS licenserna, General Public License (GPL), specificerar följande:

“When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish)./.../.”
(Open Source Initiative, 2014a. par. 3)

Walker (2013), Amaresh (2013) samt SourceForge (2014b) visar tillsammans att Wordpress, VLC, Notepad++, 7-Zip, OpenOffice och FileZilla är samtliga OSS, samtliga tillhör också de mest populära OSS genom tiderna sett till antal gånger de blivit nedladdade. En annan viktig faktor som dessa mjukvaror har gemensamt är det faktum att de är gratis.

Generellt sett är OSS gratis att införskaffa, detta kan också stödjas av att skaparna utav OSI och stora delar av open source-rörelsen i sig inte har för avsikt att kommersialisera OSS (Raymond, 2001).

Den faktiska licenskostnaden för OSS är en utgift, beroende på hur många licenser som krävs och omfattningen på mjukvaran kan denna utgift bli stor. Daffara & Gonzalez-Barahona (2000) lyfter fram att då OSS är gratis skapar detta en fördel för ett företag, då den direkta skillnaden i anförskaffningskostnad kan bli markant. Det är också denna ekonomiska skillnad som är direkt tydligast och en av de största anledningarna till att organisationer väljer att införskaffa OSS.

Det finns en tydlig skillnad när en OSS och en CSS av mindre skala jämförs, exempelvis OpenOffice mot Microsoft Office, i dessa fall är den direkta anskaffningskostnaden en påtaglig faktor (Quinn & Henri Andrei, 2013). När mjukvara av större omfattning jämförs, exempelvis Apache Webserver mot Microsoft IIS (webbserver) kan andra faktorer komma att spela roll. Vid dessa större implementeringar bör Total Cost of Ownership (TCO) övervägas (Moyle, 2004). En definition av TCO är en övergripande bedömning av ett företags IT, den inkluderar införskaffande av hårdvara och mjukvara, support, slutanvändarkostnader samt alternativkostnad för driftstopp och utbildning (Gartner, 2014) (Moyle, 2004).

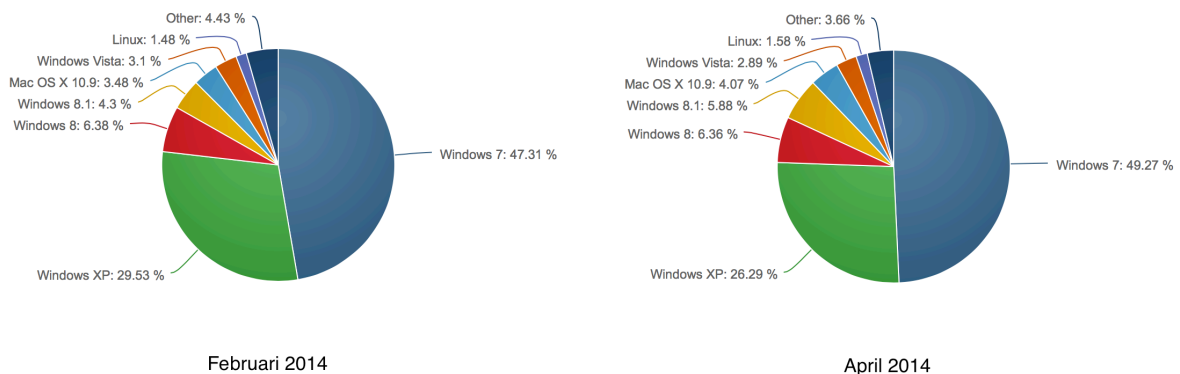
Dedrick & West (2006) genomförde en studie med 14 organisationer för att få deras syn på OSS samt i vilken utsträckning det användes. De organisationer som hade infört OSS nämnde samtliga kostnader som en faktor, och i nästan samtliga fall var detta den huvudsakliga orsaken till att införa OSS i organisationen. En del av de deltagande organisationerna hade övervägt TCO i avgörandet mellan att införa OSS eller proprietär mjukvara. För en utav organisationerna innebar det en nackdel för OSS alternativet. De omkringliggande kostnaderna för OSS alternativet, i form av underhåll och utveckling överskred helt enkelt kostnaden för det proprietära alternativet. Organisationen valde därför att överge OSS, övriga deltagande organisationer visade ett fortsatt användande eller övervägande av att använda OSS.

2.2.3 Framtidssäker investering

En utgångspunkt i huruvida en mjukvara är framtidssäker eller ej kan bestämmas av tiden som den är brukbar. Tamai & Torimitsu (1992) anser att livslängden på en mjukvara slutar när den överges av användarna och upphör då att existera. Då ingen support eller uppdateringar längre kan tillhandahållas för en mjukvara går den fortfarande att bruka men dess livslängd är passerad. När en mjukvara har nått slutet på sin livslängd av olika skäl så kan den ersättas av en nyare version eller återupplivas.

Microsoft introducerade operativsystemet Windows XP den 25 oktober år 2001, den åttonde april år 2014 tillhandahölls inte längre någon fortsatt support för systemet. Redan fem år tidigare hade ett förlängt supportprogram trätt i kraft, vilket gjorde att användare hade god tid på sig att förbereda en flytt till ett annat operativsystem. (Microsoft, 2013, 2014)

Drygt en månad innan nedstängningen av support till Windows XP, i februari 2014, genomförde analysbolaget NetMarketShare (2014) en statistisk undersökning över olika operativsystems marknadsandelar. Vid denna tidpunkt låg Windows XP på en andra plats med nästan 30 % marknadsandel, efter att datumet för nedstängning hade passerats låg marknadsandelen på 26 %, fortfarande en andra plats.



Figur 3 - Marknadsandelar operativsystem (Netmarketshare, 2014)

I en annan statistisk undersökning som genomfördes i januari år 2014 och som beskrivs av Hammond (2014), svarade 37 % utav 641 respondenter att de planerade att fortsätta använda Windows XP i sina organisationer efter nedstängningen av support för systemet. Den huvudsakliga anledningen till ett fortsatt användande uppgavs vara att Windows XP fungerar och att det då inte finns anledning att ersätta det. Andra orsaker till att inte lämna Windows XP var att andra viktiga mjukvaror är beroende av operativsystemet samt kostnaden för att ersätta systemet.

När källkoden inte är tillgänglig för användaren, vilket är fallet i proprietär mjukvara, kan support av mjukvaran inte tillhandahållas av någon annan än producenten (Tripathi & Tiwari, 2013). Eftersom att OSS har öppen källkod kan fortsatt support tillhandahållas även efter livslängdens slut, en OSS kan då överleva en proprietär motståndare (Lerner & Tirole, 2002). Även om den aktiva communityn som ger support åt ett OSS väljer att gå vidare till att arbeta med ett nytt projekt innebär detta inte slutet för den aktuella

mjukvaran. Den egna organisationen kan då fortfarande antingen själv stå för supporten genom tillgängligheten till källkoden och intern kunskap, alternativt om kunskapen saknas kan konsulttjänster införskaffas mot betalning.

Enligt Magnusson & Olsson (2005) finner organisationer en trygghet i proprietära produkter med dess supportavtal för mjukvara. Vid införandet av kritiska IT-system så som affärssystem, har i vissa fall, uteslutande stora leverantörer fått kunders förtroende. Anledningen är att det finns en trygghet i att större företag kan hållas ansvariga för själva införandet men också framåt i tiden.

En annan aspekt som belyses genom detta är det faktum att om en producent av en proprietär mjukvara försvinner från marknaden så är det svårt att förutspå vad som kommer att hända med mjukvaran. När Raymond (2001) talar om open source-utveckling i "The Cathedral and the Bazaar" ger han sin lösning på detta problem: "When you lose interest in a program, your last duty to it is to hand it off to a competent successor." (2001. s. 26). Ett bra exempel på detta är hur WordPress utvecklades från ett avbrutet open source-projekt kallat b2, tack vare den öppna källkoden kunde en ny utvecklare ta vid där skaparen avslutade (Finley, 2013).

2.2.4 Öppen källkod

Punkt två i OSI's definition av OSS specificerar följande angående källkod:

"The program must include source code, and must allow distribution in source code as well as compiled form. /.../. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. /.../."
(Open Source Initiative, 2014b. par. 2)

För att en mjukvara skall klassas som open source krävs det alltså att dess källkod görs tillgänglig. Det innebär att alla som använder en OSS har möjlighet att dissekera och analysera mjukvaran de använder, in i minsta detalj. Det är alltså fullt möjligt att undersöka kvalitén på koden, hur en specifik funktion utförs och hitta eventuella säkerhetsrisker eller buggar i mjukvaran. Transparensen som skapas i en mjukvara med tillgänglig källkod är ofta vad förespråkare av open source använder som huvudsakligt argument för fördelarna med open source (Raymond, 2001).

I en intervjuserie gjord av Dextrick & West (2006) under åren mellan 2002 och 2004, med totalt 21 anställda på 14 företag framkom det att den generella inställningen hos de 21 intervjuade var att en tillgänglig källkod är oviktig. De flesta svarade på ett sådant sätt att det framgick att de inte var intresserade av att inspektera källkoden, eller att editera den. Av de 14 företagen använde 12 antingen OSS projekten Linux eller Apache. Men endast fyra av företagen svarade att de på något sätt använde sig utav källkoden.

"It's nice to have, but having source code is like having a gun with bullets. You can shoot yourself in the foot." - Biotech's Associate Director of IT Infrastructure (Dextrick & West, 2006. s. 15).

I intervjuerna som Dextrick & West (2006) genomförde tas det även upp vad anledningen till intresse eller ointresse av källkoden var, två faktorer som togs upp var kunskap, eller brist på kunskap. Företag som hade befintlig programmeringskompetens

och anställda som kan utveckla mjukvaror var mer intresserade av källkod än de som saknade den, naturligt nog. En annan anledning var mognaden hos open source-projekt. Flera av de intervjuade som tidigare hade nyttjat källkoden och editerat denna menade att de inte längre gjorde det på grund av att open source-mjukvarorna mognat och då var på en sådan nivå att det inte längre behövdes.

2.2.5 Möjlighet att skraddarsy mjukvara

Punkt tre i organisationen OSIs definition av OSS lyder enligt följande:

"The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software." (Open Source Initiative, 2014b. par. 3)

Det innebär att en OSS får editeras, vare sig det är små eller stora modifieringar som ändrar mjukvaran i grunden. Nya editerade versioner av programvaran får även spridas under samma förutsättningar som den ursprungliga versionen (Open Source Initiative, 2014b).

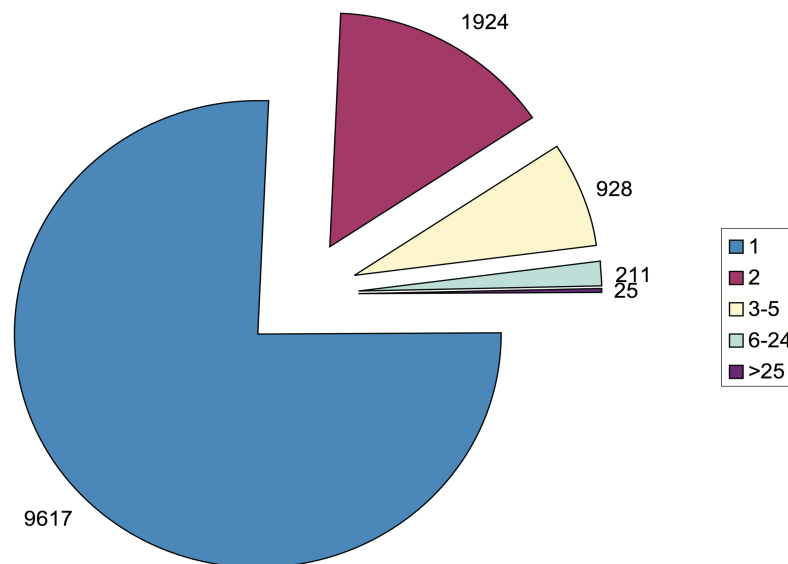
Detta möjliggör för företag, eller vanliga användare, att använda sig av en OSS och editera denna för att passa deras behov exakt. Pizka (2004) pekar på att mjukvaran även kan användas som en plattform, eller en grund, till en version som är så pass förändrad att den får klassificeras som en ny mjukvara. Detta får ställas som alternativ till att antingen låta det egna företaget utveckla en ny mjukvara från grunden, eller om kompetensen saknas, köpa tjänsten av ett annat företag. Det är dock ofta dyrt och mycket tidskrävande och bör undvikas om möjligt.

I en praktisk studie av Pizka (2004) undersöktes just möjligheten att använda en redan befintlig open source-kodbas för att modifieras efter önskemål mot att utveckla en programvarulösning från grunden. Projektet i studien spände över en tidsperiod om tio år där två typer av mjukvarulösningar utvecklades, både genom att utveckla dem från grunden och genom att använda en open source-kodbas och anpassa den. De mjukvarulösningar som utvecklades var en operativsystems-kärna (kernel) vars uppgift enkelt beskrivet är att sköta kommunikationen mellan mjukvara och hårdvara i en dator. Den andra var en kompilator (compiler) som översätter programmeringskod, av olika språk, till maskinkod som är läsbar av datorns processor. I båda dessa fall var metoden att anpassa en befintlig open source-kodbas den metod som hade kortast utvecklingstid.

2.2.6 Bidrag av och till community

En OSS förknippas ofta med en community som driver den framåt, utvecklar och förbättrar källkoden. Det är en av de största fördelarna med OSS gentemot CSS. Ju fler ögon som granskar koden desto större chans att alla buggar upptäcks, enligt *"given enough eyeballs, all bugs are shallow"* (Raymond, 2001, p. 30). Detta stöds även av de tidigare två studierna som presenteras i tidigare avsnitt 2.3.1 (tabell 1 & 3) där Lösningsgraden (ration av antalet upptäckta buggar dividerat med antalet lösta) för OSS projekt är bättre än hos motsvarande CSS. De visar dock inget gällande storlek på community eller antal bidrag av enskilda utvecklare.

En studie av Lerner & Tirole (2002) på 3149 olika open source-projekt visar att av 12 705 bidragande utvecklare så bidrog nästan $\approx 75\%$ av utvecklarna endast en gång till projektet. Figur 4 nedan visar andelen utvecklare efter antal bidrag till källkoden på totalt 3149 olika open source-projekt.



Figur 4 - Antal bidrag per utvecklare (Lerner & Tirole, 2002)

En annan studie på open source-projektet Apache av Mockus, et al. (2002) visar på en liknande trend. Där de 15 utvecklare som bidrog mest bidrog till totalt 83 % av alla modifikationsförfrågningar (Modification request), där de stod för 88 % av alla borttagna rader kod och så mycket som 91 % av alla tillagda rader kod. När det kom till buggfixar (Problem request) visades det dock att en bredare community inkluderades till en större grad, där stod de 15 utvecklarna som bidrog mest för totalt 66 % av alla buggfixar.

Även en studie av Krishnamurthy (2002) på de 100 aktiva open source-projekt som listas av webbsidan SourceForge och har givits en status "Mature" visar på att open source-projekt ofta har en relativt liten kärngrupp med utvecklare som driver projektet. Av de 100 inkluderade projekten var medelvärdet av involverade utvecklare 6.61 st. per projekt. Projektet med flest utvecklare hade 42 st. Det var dock vanligast att projekten hade en enda ensam involverad utvecklare.

Trenden pekar tydligt på att det är små kärngrupper som driver open source-projekt framåt. Lerner & Tirole (2002) uttryckte sig att *"The overall picture that we drew from our interviews and from the responses we received in reaction to the first draft of the paper is that the open source process is quite elitist"* (s. 206). Krishnamurthy (2002) beskriver deras första huvudsakliga upptäckt vara enligt följande: *"The vast majority of mature OSS programs are developed by a small number of developers"* (s. 3).

3. Metod

Kapitel tre beskriver processen för datainsamlingen till denna uppsats. Vi resonerar här kring och fastställer val av datainsamling, urval av respondenter, etiska aspekter och eventuella brister i studien.

Valet mellan en deduktiv eller en induktiv strategi uppenbarade sig vara enkelt, eftersom uppsatsens ämne är relativt invecklat och kräver en viss inblick eller kunskap från de personer som bidrar med data till empirin. Då vi ämnar undersöka huruvida företag baserar beslut angående användande av open source efter specifika faktorer behövde vi först fastställa dessa faktorer. Vi misstänkte också att det skulle bli svårt att samla empiriskt data genom intervjufrågor om vi inte ställde riktade frågor, där vi styr frågorna mot de faktorer vi upprättat. Vi valde därför att först undersöka fältet grundligt och strukturera ett ramverk av faktorer baserat på tidigare litteratur och forskning på området. För att sedan förankra empirin i det teoretiska ramverket. Uppsatsen följer alltså en deduktiv strategi för datainsamling enligt Jacobsen (2002).

3.1 Datainsamling

Valet stod mellan en kvantitativ insamling av data, i form av enkäter som sprids över Internet för att få så stort antal svar som möjligt och en kvalitativ undersökning baserad på intervjuer. Jacobsen (2002) anger svårigheten att ställa avancerade eller fördjupande frågor som en nackdel med kvantitativa undersökningar, det var något vi fann viktigt och därför föll valet på att göra en kvalitativ undersökning i form av semistrukturerade intervjuer. Sådana intervjuer möjliggör riktade och öppna frågor, på så sätt tillåts deltagarna att resonera kring ämnet och lämna detaljerade svar som ändå är inom ramen för vad vi ämnar förankra svaret till. Nackdelarna med en sådan strategi är att antalet deltagare blir drastiskt färre, och det är svårare att värja sig emot allt för subjektiva åsikter (Jacobsen, 2002). Eftersom detta inte är en undersökning som mäts i siffror eller konkret och specifik data, utan en där resonemang och fördjupande insikter från informanterna krävs, fann vi att fördelarna med en kvalitativ undersökning övervägde de i en kvantitativ.

3.1.1 Urval och informanter

Undersökningen genomfördes på fyra företag inom IT-konsultbranschen. Samtliga företag vi intervjuade arbetar med systemutveckling och mjukvaruutveckling på något sätt.

Vi valde att fokusera på denna bransch då vi fann det högst sannolikt att det inom branschen dels fanns en kunskap och insikt i vad open source är samt att vi trodde att de om någon har resonerat kring om och varför de skulle kunna tänka sig att använda open source inom verksamheten. När det kom till informanterna ville vi intervjua personer med en inblick i företagets tekniska profil, vi ville även att de skulle ha vetskap om hur administrativa beslut går till. Därför valde vi att kontakta anställda med en chefsposition på företagen.

Tabell 4 - Detaljer över intervjudeltagare

	Fishbone	Force12	Lunicore	Företag 4
Typ av företag	IT-Konsult med inriktning på Microsoft tjänster	IT-Konsult	Konsultbolag	IT-Konsult
Antal anställda	60 tal	4 st, 80 i koncern	50 tal timmanställda (2 heltid, 7 halvtid)	30 tal
Verksamma i (geografisk plats)	Göteborg, Malmö, Stockholm & Växjö	Lund	Lund	Södra sverige
Omsättning 2013 (tkr)	55 437	6530	2993 (år 2012)	>10 000
Informant	Pontus Börresen (Informant 1)	Fredrik Ljungbeck (Informant 2)	Elias Gabriellsson (Informant 3)	Anonym (Informant 4)
Befattning	Regionchef södra Sverige	Systemutvecklare	Affärsområdeschef för IT	Teknikchef

3.1.2 Intervjuguide

Frågorna (Bilaga 1) utformades för att kunna kopplas till de sex faktorer som fastställs i det teoretiska ramverket (Kapitel 2). Men eftersom målet var att undersöka huruvida företagens syn på ämnet stämmer överens med de faktorer vi tagit fram ville vi inte ställa frågor där frågan för starkt förknippas med bara ett karaktärsdrag hos open source eller closed source. Vi ville exempelvis undvika frågor likt: "När ni implementerar nya mjukvaror, är licenskostnaden en stor faktor i beslutet?" en sådan fråga blir för tydligt kopplad till just en av våra faktorer, och vi ansåg att svaret på en sådan fråga förmodligen skulle bli isolerat från övriga faktorer.

Intervjuerna genomfördes på en plats föreslagen av informanterna, vilket förutom med Lunicore blev på företagets kontor. Intervjun med Lunicore genomfördes istället på en avskild del av ett café. Eftersom vi är två författare valde vi att ha en intervjuledare per intervju. Intervjuledaren styrde intervjun och ställde alla frågor, den andra tog anteckningar och inflikte endast i intervjun om denna fann att intervjuledaren missade något av vikt. Intervjuerna inleddes med att be om tillåtelse att ta ljudupptagning av intervjun, på så sätt tilläts ett flyt i intervjun utan att pauser för att hinna med att anteckna. Vidare frågade vi om informanten ville vara anonym personligen, eller om denne ville anonymisera företaget informanten representerade.

För att vi skulle kunna försäkra oss om att informanten förstod och hade ungefär samma uppfattning om vad open source är, begreppet och karaktärsdragen är komplexa, började vi alla intervjuer med frågan: "Vilken är din tolkning av begreppet Open Source Software?" varefter vi, om deras uppfattning inte stämde överens med vår, presenterade en förenklad definition av open source enligt följande.

- Får distribueras utan begränsningar
- Källkoden är fritt tillgänglig
- Det är fritt fram att editera källkoden och distribuera nya versioner.

Vi ville ställa frågor där informanten gavs rum att inte bara tänka på ett karaktärsdrag hos open source eller closed source.

3.2 Analys

Jacobsen (2002) beskriver analysprocessen av kvalitativ data i tre steg; beskrivning, systematisering & kategorisering samt kombination. Dessa tre faser används för att reducera insamlad data till dess minsta väsentliga beståndsdelar. Första steget efter att intervjuerna genomförts var att transkribera ljudupptagningarna för att underlätta arbetet med att få en överblick av materialet. Vid transkriberingen inkluderades inte eventuellt kallprat och oväsentliga diskussioner, som ofta uppstår innan och efter själva intervjun. Ur transkriberingarna presenterades sedan allt material av vikt kategoriserat efter de tre kategorier som upprättas i det teoretiska ramverket, det för att ge en tydlig översikt vilka svar som hör till vilka faktorer från ramverket.

3.3 Etik

Vi hade från start Jacobsens (2002) tre etiska aspekter informerat samtycke, krav på privatliv och krav på att bli korrekt återgiven i åtanke under hela intervjuprocessen. Vid såväl utformning av intervjun, i kontakten med informanter, under intervjun och i analysen av intervjun. Ett informerat samtycke innebär att informanten deltar av fri vilja, en förutsättning för det är att denna är medveten om uppsatsens och intervjuernas syfte i såpass stor utsträckning det är möjligt. För att försäkra oss om detta gav vi en kortare beskrivning av uppsatsens ämne och mål till informanterna vid första kontakttillfället, vi erbjöd även att i förväg skicka intervjufrågorna och ge såväl företaget som informanten anonymitet. Faktorn krav på privatliv menar Jacobsen (2002) är viktigast när ämnet är av en natur som närmar sig informantens privata sfär. Denna undersökning har ingen sådan karaktär, utan riktar sig mot företagen. Även där kan det dock finnas känsliga ämnen och information, därav erbjudandet om anonymitet. Till sist är det viktigt att de insamlade svaren blir korrekt återgivna. Fullständig återgivning är inte möjlig, istället är det viktigt att svaren redovisas fullständigt där det är möjligt och att inte lösryckt plocka fördelaktiga svar eller resultat från empirin. Jacobsen (2002) nämner även vikten av att inte förfälska eller förvränga resultat. Detta är något vi tog på största allvar, vi transkriberade därför intervjuerna ord för ord så fort som möjligt efter intervjutillfället och skickade dem till informanterna med instruktioner att återkomma om de inte fann dem korrekt återgivna.

3.4 Kritik av metodval

Den kanske mest uppenbara svagheten med en kvalitativ undersökning är, precis som Jacobsen (2002) skriver att på grund av formatet går det inte att samla in data från ett lika stort antal informanter som vid en kvantitativ undersökning. Det innebär att det blir svårare att generalisera (Jacobsen, 2002). Vi gjorde fyra intervjuer, det ger inte tillräckligt stor bredd av data för att kunna generalisera på området.

4. Empiri

I kapitel fyra presenteras resultatet av den empiriska undersökningen. De fyra intervjuerna sammanställs och intervju svaren kategoriseras efter de tre områden i det teoretiska ramverket de tillhör.

Tabell 5 - Deltagande informanter och företag

	Fishbone	Force12	Lunicore	Företag 4
Informant	Informant 1	Informant 2	Informant 3	Informant 4

En inledande analys av intervjuresultatet presenterat under respektive kategori från ramverket ges även i detta kapitel, det innebär att våra resonemang kommer att presenteras kortfattat för att sedan i kapitel fem ges mer utrymme och utvecklas vidare.

Intervjufrågorna består av två karaktärer, dels generella frågor som är tänkta att ge en inblick i informantens syn på open source samt företagets inställning till fria mjukvaror. Svaren på dessa frågor behandlas i underkapitel 4.2.

Den andra karaktären av frågor är de frågor som är till för att förankras med någon av de tre kategorierna i det teoretiska ramverket, och dess sex faktorer. Dessa frågor och svaren på dem presenteras i respektive underkapitel i 4.3 uppdelat efter de tre kategorierna "Säkerhet", "Ekonomi" och "Utveckling" i ramverket.

Varje intervju avslutades med att informanterna bads rangordna sju faktorer, där ett är viktigast och sju är minst viktigast. De faktorer vi bad informanterna rangordna är, även om de inte kallas vid samma namn direkt kopplade till de sex faktorerna i det teoretiska ramverket (Kapitel 2.1). Nedan följer tabell 6 som visar deras relation.

Tabell 6 - Relation mellan faktorer i intervju och teoretiskt ramverk

Intervjufaktorer	Faktorer i teoretiskt ramverk	Kategori i teoretiskt ramverk
Kvalitet	Grad av buggar och sårbarheter	Säkerhet
Säkerhet	Grad av buggar och sårbarheter	Säkerhet
Låg kostnad	Licenskostnad & supportkostnad	Ekonomi
Framtidssäkerhet	Framtidssäker investering	Ekonomi
Tillgång till källkod	Öppen källkod	Utveckling
Anpassningsbar lösning (Skräddarsydd)	Möjlighet att skräddarsy mjukvara	Utveckling
Support	Bidrag av och till community	Utveckling

Sammanställningen av informanternas svar, samt ett medelvärde på faktorn illustreras i tabell 7 nedan.

Tabell 7 - Rangordnade faktorer av informanter

	Fishbone	Force12	Lunicore	Företag 4	Medelvärde
Kvalitet	1	3	3	1	2
Låg kostnad	7	6	1	2	4
Säkerhet	5	2	5	4	4
Tillgång till källkod	6	7	6	7	6,5
Anpassningsbar lösning (Skräddarsydd)	3	4	4	6	4,25
Support	4	5	7	5	5,25
Framtidssäkerhet	2	1	2	3	2

4.1 Presentation av intervjuer

Fishbone är ett företag som är aktivt på flera orter i Sverige. Informant 1 som intervjuades är dock koncentrerad till regionen södra Sverige. De kategoriserar sig som ett IT-konsultbolag med inriktning på Microsofts tjänster och produkter. De är även Microsoft gold partners inom tre områden.

Lunicore är ett konsultbolag verksamt i Lund med omnejd, med en inriktning mot It & Management. Företaget är uppbyggt och drivs till fullo av studenter på Lunds Universitet. De utvecklar en mängd olika mjukvarulösningar i varierande storlek efter deras kunders önskemål.

Force12 är ett IT-konsultbolag som ingår i Avensia-koncernen. Företaget är verksamt i Lund med omnejd. De arbetar med systemutveckling inriktat på Microsofts miljö.

Det fjärde företaget önskade att vara anonyma, av den anledningen följer endast en förenklad beskrivning. Företaget är verksamma i södra Sverige och kategoriseras som ett IT-konsult bolag med inriktning på Microsofts tjänster. De utvecklar en rad olika produkter efter deras kunders önskemål och har även förvaltning av dessa system inom deras verksamhet.

4.2 Användning och uppfattning av OSS

Varje intervju inleddes med att informanterna ombads definiera deras uppfattning om vad begreppet open source innebär. Vår uppfattning av open source hade vi reducerat till tre punkter, får distribueras utan begränsningar, källkoden är fullt tillgänglig och att det är fritt fram att editera källkoden och distribuera nya versioner. Denna presenterades till alla informanter där vi fann att deras tolkning inte stämde överens med vår.

Informant 1 tolkade open source som:

*“olicensierad, fri för alla att både använda och framförallt bidra till.”
(Informant 1, 2014).*

Informant 2 beskrev det något tveksamt som tillgång till källkod som dessutom går att editera på egen hand.

Informant 3 avvek från de andra och tog upp aspekter som att det finns ett community som möjliggör att flera utvecklare deltar

“När jag hör open source så tänker jag kanske på två saker, det första är när det finns communitys som utvecklar. Alltså det finns enskilda utvecklare som har ett intresse av vissa saker då, det är en utbytesform som möjliggör utveckling tillsammans över flera individer /.../.” (Informant 3, 2014).

Sedan nämnde Informant 3 även transparens och standardisering i sin definition.

Informant 4 gav det utförligaste svaret och tog upp faktorer som att källkoden distribueras publikt, vem som helst tillåts att editera och skapa egna versioner. Informant 4 utvecklade även ytterligare med att förklara att OSS inte nödvändigtvis behöver vara men oftast är gratis.

På frågan om anställda på företaget fritt får installera de programvaror de vill på sina arbetsdatorer svarade alla att de anställda hade frihet att göra det.

På frågan om OSS användes inom verksamheten, som en standardprogramvara, svarade endast en, Fishbone, att de definitivt inte gjorde det. De övriga svaren var varierande, Lunicore använder inte open source bland deras centrala standardprogram, de använder dock open source-webbläsare etc. på personligt plan. Det skall dock tilläggas att de i deras utvecklingsarbete använder flertalet utvecklingsbibliotek och programmeringsspråk som är open source. Även Force12 svarade att de använder open source-utvecklingsbibliotek och programmeringsspråk i deras utvecklingsarbete men programvarorna de använder i arbetet är alla CSS.

Informant 4 svarade däremot att företag 4 förmodligen använder open source, programvarorna Sublime Text och Notepad++ togs upp som exempel. Efter intervjun undersöktes mjukvarorna som nämnts, och mycket riktigt visar det sig att Notepad++ är en OSS.

4.3 Svar efter kategori i ramverk

4.3.1 Säkerhet

Tabell 8 - Rangordnande faktorer i säkerhetskategorin

	Fishbone	Force12	Lunicore	Företag 4	Medelvärde
Säkerhet	5	2	5	4	4
Kvalitet	1	3	3	1	2

Faktorn "Säkerhet" rangordnas generellt på en låg nivå av de intervjuade företagen. I samband med att informant 4 gjorde sin rangordning av faktorerna svarade informanten:

"Jag kryssade för säkerhet för att det kan ju vara känslig data vi jobbar med och då är det bra om man kan förlita sig på produkten." (Informant 4, 2014)

Endast Force12 valde att rangordna "Säkerhet" i topp, dock skulle detta kunna bero på att informant 2 tolkade faktorn annorlunda. Anledningen till det är att på en fråga som ställdes tidigare i intervjun huruvida Force12 har en bild av att dem ska jobba med typen av mjukvaror som är bundna till ett stort företag som Microsoft svarade Informant 2 att:

"/.../. När man själv är lite kan man då ta rygg och rida på ett större varumärke som vi gjorde på Microsoft. /.../. Vi kan hjälpa er men skulle vi falla eller någonting så är det ändå en standardplattform." (Informant 2, 2014)

Det leder oss till att tro att Informant 2 tolkade "Säkerhet" som säkerhet i form av trygghet och inte enligt den tolkning som ligger i linje med vår faktor "Grad av buggar och sårbarheter".

Intervjufrågan som är inriktad på säkerhet ställdes endast till de företag som idag använder OSS i sin verksamhet. Frågan undersöker om företaget gör någon form av analys på programvaran, på källkoden exempelvis, innan en OSS implementeras. Både Lunicore och företag 4 säger att de inte analyserar programvaran innan implementering, åtminstone inte i utgångsläget som ett formellt tillvägagångsätt.

Anledningen till att inte utföra en analys av programvaran skiljer sig något åt mellan företagen. I Lunicores fall så beror det på att det inte finns resurser till att genomföra en sådan uppgift.

"Vi brukar ju inte göra reviews på källkoden själv för det har vi inte resurser till /.../." (Informant 3, 2014)

Medan det i företag 4's fall beror på att det mest används programvara som använts sedan tidigare.

Dock säger både Lunicore och företag 4 att det finns undantag. För företag 4 kan en mindre omfattande analys genomföras om en programvara som är helt otestad behövs i

verksamheten. Lunicore gör sällan en analys av källkoden, däremot kan andra aspekter av en programvara analyseras, exempelvis när större system eftersöks.

"/.../. Däremot så vet jag att vi i fall där vi har behövt utvärdera tyngre system så har vi kollat på frameworket som helhet. Då är det snarare kanske inte att den är just open source, men att hur sker maintenance av den här mjukvaran och vad har den för managementsätt kring den." (Informant 3, 2014)

"Kvalitet" är tillsammans med "Framtidssäkerhet" den faktor som värdesätts högst av alla faktorer, den anses vara viktig av alla informanter, och som tabell 6 visar absolut viktigaste av två utav dem.

På frågan huruvida företagen generellt sett tror att open source-mjukvaror har fler eller färre buggar och fel än motsvarande closed source-mjukvara svarade samtliga informanter snarlikt varandra. Endast informant 2 tror att OSS generellt sett innehåller färre buggar och fel än motsvarande CSS.

"jag kan nog.. känslan är att det är färre men jag tror att det är en snabbare åtgärdstid i alla fall. Det är bara min känsla." (Informant 2, 2014)

Informant 1 tror att OSS generellt innehöll fler buggar förr i tiden, men att så inte är fallet nu.

"Men jag tror inte att det finns fler buggar, förr kanske det var så. Men jag tror inte att det finns det nu." (Informant 1, 2014)

Informant 4 poängterar att det finns säkerhetsbrister hos såväl CSS som OSS, där en högst aktuell säkerhetsbrist hos OSS nämns.

"Jag har använt många program som varit closed source som har skitjobbiga buggar och det finns ju open source som har farliga buggar också, typ HeartBleed." (Informant 4, 2014)

Alla informanter utom informant 1 tror att buggar och fel hittas snabbare i OSS, informant 2 och 4 tror också att när väl buggar och fel upptäcks i OSS så åtgärdas de snabbare.

"Det finns ju fördelar med open source som att det finns många som kanske bidrar och då kan hitta buggar och när de hittas buggar kanske de fixas snabbare." (Informant 4, 2014)

Informant 3 hade däremot en annorlunda syn på detta i relation till de andra informanterna. Informanten svarade att det kanske inte är huruvida mjukvaran är open source eller closed source utan snarare att utvecklingsarbetet är testdrivet eller ej som är avgörande i kvalitén på mjukvaran.

" /.../här får man snarare titta på sättet man hanterar utvecklingen på än om den är open source eller closed source. Om det är så att man har en väldigt testdriven utveckling med tydligt definierat hur man hanterar fel, det tror jag snarare motverkar felet än om den är open source eller closed source" (Informant 3, 2014)

4.3.2 Ekonomi

Tabell 9 - Rangordnade faktorer i ekonomikategorin

	Fishbone	Force12	Lunicore	Företag 4	Medelvärde
Låg kostnad	7	6	1	2	4
Framtidssäkerhet	2	1	2	3	2

När de intervjuade företagen ombads att ranka en lista med vilka faktorer som är viktigast för dem vid införandet utav ny mjukvara skiljer sig svaren påtagligt åt.

Faktorn "Låg kostnad" placeras genomsnittligt på en delad andra plats av faktorer. "Säkerhet" anses vara av lika hög vikt som en låg kostnad, tätt följt utav "Anpassningsbar lösning". Skillnaden mellan hur de olika företagen har värderat "Låg kostnad" som en avgörande faktor är dock stor, där två utav företagen har placerat denna faktor i topp har de andra två företagen placerat faktorn i botten av listan. Faktorn "Framtidssäkerhet" rangordnas genomsnittligt på en delad första plats av faktorer, "Kvalitet" i mjukvaran anses genomsnittligt vara lika viktigt för de intervjuade företagen.

På frågan om vilken som var den huvudsakliga orsaken för företagen att använda open source skiljer sig svaren en del åt. Hälften av företagen använde OSS i sin verksamhet, även om det inte var en standardprogramvara, och av dessa var det ett företag, företag 4, som angav "Låg kostnad" som en utav de bidragande faktorerna. Detta var dock inte den huvudsakliga orsaken, där snarare den bidragande orsaken var att den mjukvara som användes fungerade bra samt underlättade arbetet.

Fishbone hade inte infört OSS i verksamheten, då detta företag har ett partnerskap med Microsoft av guldstatus. Detta innebär att företaget får använda Microsofts produkter som omfattas av partnerskapet, utan kostnad.

På frågan huruvida Fishbone hade haft open source-mjukvaror i åtanke överhuvudtaget gav informant 1 svaret:

"Jo men det har vi, vi utvärderar ju. Men jag tror det blir lite så per automatik att okej; kan vi ta någonting som för andra kostar då litegrann, det är nog superbra, då kör vi det, för oss är det gratis. Man utvärderar nog inte alla alternativ, tror jag." (Informant 1, 2014).

Detta visar på att det finns en kostnadsmedvetenhet som en bidragande faktor för att välja en viss typ av mjukvara. Övriga informanter nämnde aldrig några ekonomiska för eller nackdelar med de olika alternativen av mjukvara under den semistrukturerade delen av intervjun.

Samtliga intervjuade företag ser "Framtidssäkerhet" som en viktig faktor vid införandet av ny mjukvara i sin organisation. Vad som skapar en framtidssäker mjukvara ter sig dock olika mellan de intervjuade företagen, där både Force12 och Fishbone ser en säkerhet i proprietär mjukvara, i synnerhet Microsofts alternativ.

"/.../. Dem är ju mycket att de lägger fram 5 eller 10 års planer, då går dem på det säkra kortet, med Microsoft som de vet att det finns support i 10 år. De vet att det finns nya versioner som kommer. För dem klarar inte av att svänga sin verksamhet lika snabbt som, i alla fall är det min bild att en open source

*mjukvara kan ändra riktning lite fortare för det är vi som påverkar den./.../.”
(Informant 1, 2014)*

Medan Lunicore ser en framtidssäkerhet i OSS där transparensen i källkoden är lika viktig som det faktum att en bindning mot ett tredjepartsföretag kan undvikas.

“/.../. I med att vårt bolag som sådant inte är jättestort så måste vi lova våra kunder att det vi utvecklar har en lång livslängd och genom att förankra det i någonting som open source så vet vi att det finns andra aktörer som också skulle kunna plocka upp andra projekt som vi har levererat. Utan att det skulle innefatta en ekonomisk bindning mot ett tredjepartsföretag som har sin egen lösning.” (Informant 3, 2014)

4.3.3 Utveckling

Tabell 10 - Rangordnade faktorer i utvecklingskategorin

	Fishbone	Force12	Lunicore	Företag 4	Medelvärde
Tillgång till källkod	6	7	6	7	6,5
Anpassningsbar lösning (Skräddarsydd)	3	4	4	6	4,25
Support	4	5	7	5	5,25

De tre faktorerna i utvecklingskategorin visar på en tydlig trend bland de intervjuade företagen att de karaktärsdrag hos open source som är kopplade till utveckling är av mindre vikt för dem. Det är faktiskt så lågt prioriterat att de tre faktorerna är de minst viktiga faktorerna av alla, där ”Tillgång till källkod” är absolut lägst värderad.

Det stöds även av de semistrukturerade frågorna kopplade till utvecklingskategorin. Endast två av företagen använder OSS, och de använder det dessutom i mindre utsträckning. En förutsättning för att dessa faktorer skall värderas högt är att open source används, då de inte ens möjliggörs av en proprietär mjukvara.

Av de två företag som använder OSS var det endast Företag 4 som över huvud taget utnyttjat den öppna källkoden i ett open source-projekt. Där informant 4 svarade att de vid någon enstaka gång editerat en open source-lösning för att skräddarsy den efter deras behov. Informant 3 berättar att de inte gör några analyser på eller editerar någon källkod, däremot nämner informanten fördelarna öppen källkod skapar för dem, på en följdfråga om transparensen i open source är viktigt för dem svarade Informant 3 att:

”Precis och också för att vi ska kunna lova våra, i med att vårt bolag som sådant inte är jättestort så måste vi lova våra kunder att det vi utvecklar har en lång livslängd och genom att förankra det i någonting som open source så vet vi att det finns andra aktörer som också skulle kunna plocka upp andra projekt som vi har levererat. Utan att det skulle innefatta en ekonomisk bindning mot ett tredjepartsföretag som har sin egen lösning.” (Informant 3, 2014)

5. Diskussion

I kapitel fem analyseras empirin från kapitel fyra ytterligare och diskuteras i relation till det teoretiska ramverket och tidigare forskning som presenteras i kapitel två. Våra resonemang kommer att framföras och presenteras efter den kategori från det teoretiska ramverket de tillhör, säkerhet, ekonomi eller utveckling.

Alla dem vi intervjuade var positivt inställda till open source och fann inga problem med kvalitén eller säkerhet hos open source. Dock var det ingen av dem som använder open source i någon större skala, endast två av dem använder enstaka mindre mjukvaror. Det är intressant då det inte följer den trend som rapporterats av exempelvis Gartner (2011) där open source används av cirka hälften av de deltagande företagen och där ytterligare cirka 20 % planerar en implementation av OSS. Det kan finnas flera anledningar till att situationen ser ut som den gör bland de deltagande företagen, det går inte att avgöra om det är ett generellt resultat med endast fyra respondenter. Vi finner dock två troliga anledningar. Den första är att de små till medelstora företag vi intervjuat finner en trygghet i att använda mjukvara av proprietär sort på grund av säkerheten det ger, dels i och med att de internt vet vad de får och att det finns support att få etc. men även att deras kunder finner en säkerhet i att dessa mindre företag arbetar med standardiserad mjukvara, som Informant 2 förklarade. Den andra faktorn som inte får bortses ifrån är att flera av företagen levererar produkter med inriktning mot Microsoft, som är en leverantör av proprietär mjukvara. Därav underlättar det att hålla sig fast vid deras produkter.

Säkerhet

Informanternas uppfattning om säkerheten i OSS är positiv, ingen av de intervjuade trodde att OSS hade svagare säkerhet än CSS. Istället trodde de att den var likvärdig eller till och med bättre. Denna uppfattning stämmer bra överens med verkligheten sett till undersökningarna gjorda av Schryen & Rich (2010) och Mockus et al. (2000). De visar båda två att OSS och CSS är likvärdiga när det kommer till antal buggar och sårbarheter.

En av informanterna nämnde även att denne hade uppfattningen att fel och buggar hittas och rättas till fortare hos OSS än hos CSS. Även den uppfattningen kan styrkas av undersökningarna gjorda av Schryen & Rich (2010) och Mockus et al. (2000).

Att fel och buggar hittas och rättas till fortare är något som många förespråkare till OSS hävdar vara en av de stora fördelarna med open source, exempelvis är Raymond (2001) en av dem. Det kan låta logiskt att många utvecklare är bättre än få, men det är viktigt att inte glömma att i proprietär mjukvaruutveckling finns ett vinstintresse. Med det kommer en vilja att effektivisera, och med det en drivkraft att standardisera och strukturera utvecklingsprocessen. Många utvecklare må vara bra, men om de är oorganiserade alternativt oerfarna eller okunniga är de inte mycket till hjälp. Undersökningarna av Schryen & Rich (2010) och Mockus et al. (2000) visar på en trend att open source-projekt har en mycket högre grad fel och buggar när mjukvaran släpps till allmänheten än vad proprietär mjukvara har. Open source-projekt släpps ofta i en mycket tidigare version än vad closed source gör, i så kallade alpha eller beta versioner. Att graden av fel och buggar samt lösningsgraden på dem är mycket högre i början är därför föga förvånande. När open source-projekten mognar tenderar de oftast hamna på

ungefär samma nivå som closed source motsvarigheten, eller bättre (Schryen & Rich, 2010) (Mockus et al., 2000).

Ekonomi

Tidigare forskning visar att låg kostnad är den huvudsakliga orsaken till att välja ett OSS alternativ framför en CSS. Detta överensstämmer inte väl med de svar som studiens informanter gav. Hälften av informanterna angav "Låg kostnad" som den viktigaste eller näst viktigaste faktorn när det kommer till valet av mjukvara. Detta går delvis emot Dedrick & West (2006) forskning som visar att samtliga i studien deltagande företag angav kostnad som en faktor för att införa OSS i verksamheten, och nästan samtliga företag angav denna faktor som huvudsaklig anledning.

Daffara & Gonzalez-Barahona (2000) nämner att skillnaden i direkt anskaffningskostnad kan bli markant i valet mellan en OSS och en CSS, och att det är denna ekonomiska skillnad som är en av de största anledningarna för en organisation att välja ett OSS alternativ. Quinn & Henri Andrei (2013) menar att den direkta anskaffningskostnaden är en påtaglig faktor när en OSS och en CSS av mindre skala jämförs.

Det är mindre mjukvaror som används utav företagen, vilket då bör vara kopplat till att dessa mjukvaror är gratis och därför fördelaktiga att användas i jämförelse med deras proprietära motsvarighet, men så är inte fallet. Att företagen inte värderar den låga kostnaden med OSS högt tror vi kunna bero på företagens storlek.

Två av studiens deltagande företag är s.k. Microsoft gold partner vilket innebär att de inte behöver betala för att använda deras produkter. Kostnadsmedvetenheten finns där, eftersom de ser en fördel med att kunna använda Microsofts produkter utan kostnad. Det infinner sig dock aldrig någon riktig anledning att utmana de proprietära alternativen för dessa företag. Vi tror att detta beror på företagens storlek men också att de är inriktade på att arbeta i Microsofts miljö. Kan en beprövad och erkänt kvalitativ mjukvara användas gratis, då finns det ingen anledning att ifrågasätta alternativet.

Inget utav de intervjuade företagen utför en analys av TCO i samband med val av mjukvara. Enligt Moyle (2004) bör detta utföras i samband med att mjukvara av större omfattning ska implementeras i organisationen. Att de intervjuade företagen väljer att inte genomföra denna analys tror vi även det kan bero på deras storlek, det finns helt enkelt inte tid och resurser att utföra ett omfattande analysarbete.

Huruvida en mjukvara är framtidssäker eller ej anses vara av hög vikt för de i studien deltagande företagen. Detta går att sammankoppla med en statistisk undersökning beskriven av Hammond (2014) som pekar på att företag tenderar att hålla fast vid en mjukvara så länge som bara möjligt.

Det finns en tydlig skillnad mellan de deltagande företagens syn på framtidssäkerhet, företagen med en tydlig profilering mot Microsofts miljö har en bild medan företagen som har tagit till sig OSS har en annan bild. Det finns inga undersökningar som visar på att bilden ska skilja sig, däremot finns det tidigare utförd forskning som visar på tydliga fördelar för respektive sida. Magnusson & Olsson (2005) talar om en trygghet som går att finna i de proprietära alternativen med dess tillhörande supportavtal. Större leverantörer uppges inge ett större förtroende hos kunder i många fall. Detta stämmer överens med hur informant 1 talar om framtidssäkerhet i mjukvara. Den garanti för vad

som komma skall, som upplevs finnas i proprietär mjukvara, skapar en trygghet som OSS inte kan ge dessa företag. Enligt Lerner & Tirole (2002) kan öppen källkod i en mjukvara innebära en möjlighet att skapa support, även efter att mjukvarans livslängd nått sitt slut. Detta skulle teoretiskt sett kunna innebära att ett OSS kan överleva ett proprietärt alternativ, dels för att supporten inte är tidsbegränsad men även för att ett tidigare avslutat open source-projekt kan återupplivas. Informant 3 bekräftar att transparensen i källkoden skapar en framtidssäkerhet, där även en ekonomisk bindning till ett tredjepartsföretag kan undvikas.

Vi tror att IT-företag kan finna en säkerhet och trygghet i att välja en stor aktör som leverantör av mjukvara, det finns då alltid någon att hålla till svars, någon som är ansvarig framåt i tiden. Med ett partnerskap till en stor leverantör som Microsoft blir användandet av ny proprietär mjukvara ingen ny investering, det tillhandhålls utan kostnad som en del av partnerskapet. Det verkar då inte finnas någon ekonomisk vinning i att ifrågasätta sitt tillvägagångssätt. Om ett företag däremot inte är inriktat på att arbeta i Microsofts miljö så kan OSS innebära en mer attraktiv framtidssäkerhet, där transparensen i källkoden försäkrar en mjukvara med lång livslängd, och en ekonomisk bindning mot en leverantör undviks.

Utveckling

Denna kategori visade sig vara minst viktig för de fyra intervjuade företagen. De tre faktorerna "Öppen källkod", "Möjlighet att skraddarsy mjukvara" och "Bidrag av och till community" var de tre minst värdesatta faktorerna överlag. Inledningsvis var vi något ställda över detta resultat då företagen vi intervjuade alla är IT-konsultbolag, vilket innebär att de alla arbetar med mjukvaruutveckling. De har alltså intern kunskap att analysera källkoden, anpassa den efter deras behov och bidra tillbaka till communityn. Men ingen av de två som använder open source nyttjar de möjligheterna i någon större utsträckning, och prioriterar det lågt precis som de två som inte använder open source alls. En av de intervjuade svarade att det inte fanns resurser till det, och det kan nog vara ett rimligt antagande att det inte är värt att spendera tid och resurser på att analysera källkod eller att modifiera och skraddarsy mjukvaror på annat än stora företag. Dessa små mjukvaror som används av företagen vi intervjuat har dessutom väldigt specifika användningsområden och det finns ingen funktionalitet i dem som kräver att de anpassas efter företaget. Det är troligare att det blir intressantare att både granska och editera källkoden om det är en omfattande mjukvara där företaget kan tänkas ha önskemål om skraddarsydda funktioner, ett Content management eller Customer relationship system kan tänkas vara sådan mjukvara.

Det är anmärkningsvärt att inga av företagen har något större intresse i faktorn "Öppen källkod", som får anses vara själva essensen av open source. Det är den öppna källkoden som skapar alla skillnader mellan open source och proprietär mjukvara. Men ändå finns det knappt något intresse av det bland de intervjuade företagen. Även om företaget inte vill använda just en open source-mjukvara inom sin egen miljö så arbetar alla deltagande företag med mjukvaruutveckling. Det borde rimligtvis finnas värde i att analysera befintliga OSS för att ta lärdom och inspiration ifrån, eller till och med använda delar av open source-produkter i deras egna lösningar, som i studien av Pizka (2004) visar sig vara mycket effektivt. Att intresset för detta är så lågt att endast en av fyra över huvud taget nämner att de tänkt på att bidra till open source communityn är förvånande. Vi tror att även detta beror på, som nämnts tidigare, företagets storlek.

Sammanfattning

Vi valde att inkludera företag av lite mindre storlek och inom IT-konsultbranschen i studien för att vi fann det troligt att de skulle ha resonerat kring open source och även använda det på grund av de ekonomiska fördelarna det kan innebära. Det visade sig att de alla hade bra resonemang kring open source och hade haft det i åtanke, dock var det egentligen inte någon som använde det i den utsträckning vi hade förväntat oss. Detta trots att open source sågs likvärdigt eller till och med bättre kvalitetsmässigt av samtliga deltagande företag. Det finns alltså en medvetenhet om open source, de tror att open source håller hög kvalitet, de anser open source vara minst lika säkert, de har en prisedvetenhet och rankar "Låg kostnad" som bland det viktigaste. De arbetar alla med mjukvaruutveckling vilket borde innebära att det finns ett värde av öppen källkod. Trots detta är det ingen av dem som använder det i mer än en mycket begränsad utsträckning. De är alltså mer intresserade av en mjukvarumodell som erbjuds av "Freeware" mjukvaror.

6. Slutsats

I kapitel sex presenteras uppsatsens resultat.

Syftet med denna uppsats var att kartlägga vilka faktorer som IT-företag resonerar kring när de implementerar open source mjukvara. Den huvudsakliga forskningsfrågan var därför: *Vilka faktorer är avgörande när IT-företag väljer en open source-mjukvara framför de proprietära alternativen?*

Intervjuerna med företagen visar tydligt på att de sex faktorer det teoretiska ramverket innehåller är sådant som dessa företag resonerar kring när de kommer till val av mjukvaror. Såväl de semistrukturerade frågorna som frågan där de ombads rangordna faktorer visade att informanterna hade resonemang kring ämnet på såväl open source som closed source. Alla hade en generellt positiv inställning till open source, ingen av dem nämnde att open source skulle vara svagare än proprietära mjukvaror på någon punkt. Informanterna hade en bild av att säkerheten i OSS var likvärdig eller bättre än hos motsvarande CSS, det nämndes även i ett fall att fel och buggar hittas och korrigeras snabbare hos OSS än hos CSS. Denna uppfattning styrks av tidigare genomförda studier, av exempelvis Schryen & Rich (2010), vi finner det därför rimligt att säkerhetsaspekten inte rangordnas speciellt högt utan ges ett medelhögt värde.

Något som framkom tydligt var företagens ointresse för utvecklingsmöjligheterna hos open source. "Öppen källkod", "Möjlighet att skraddarsy mjukvara" och "Bidrag av och till community" som alla tillhör utvecklingskategorin i vårt ramverk var de faktorer som var minst intressanta överlag, men även om de inte var av intresse för respondenterna kunde de ändå resonera kring dess fördelar och nackdelar på ett övertygande sätt. Att dessa faktorer rangordnades så lågt av såväl företagen i vår studie och i andra studier av exempelvis Dedrick & West (2006) förvånade oss. Den öppna källkoden i open source anser vi vara den största skillnaden på open source och closed source, vi hade därför på förhand antagit att det skulle vara utvecklingskategorin och framförallt faktorn "Öppen källkod" som ansågs vara viktigast. Vi finner det även förvånansvärt att ingen av de deltagande företagen sätter ett värde på att analysera existerande OSS för att kunna dra lärdom eller få inspiration ifrån dessa, det borde rimligtvis vara åtråvärt att kunna använda delar av lösningarna till den egna befintliga verksamheten. Vi drar slutsatsen att detta beror på företagens småskaliga storlek med få resurser för att kunna genomföra denna typ av aktioner.

De faktorer som var viktigast överlag var istället "Framtidssäker investering" och "Kvalitetsfokus". Faktorn "Licens & Supportkostnad" var antingen bland det viktigaste eller inte viktigt alls. Att båda faktorerna i ekonomikategorin "Framtidssäker investering" och "Licens & Supportkostnad" värderades så pass högt, samtidigt som utvecklingskategorin värderades så pass lågt tyder på att de inte är speciellt intresserad av mjukvarumodellen open source levererar utan mer åt en Freeware-modell.

Uppsatsen undersökte även i vilken utsträckning IT-företag idag kan tänka sig att använda open source över huvud taget, en andra forskningsfråga blev därför: *Är open source över huvud taget ett reellt alternativ för dessa företag?*

På förhand hade vi väntat oss att alla dessa små till medelstora IT-företag vi intervjuade skulle använda sig av någon form av OSS bland deras standardprogramvaror i verksamheten. Det visade sig istället vara tvärt om, inget av de intervjuade företagen använde OSS på det sätt vi förväntat oss. Någon använde en eller ett par enskilda mindre mjukvaror som de antog vara open source (vilket de visade sig vara), men de var inte en standardprogramvara i verksamheten. Flera använde open source-bibliotek i deras utvecklingsarbete, alltså användes det till deras produkter de levererar externt men inte internt. Detta resultat stämmer inte överens med vad som rapporterats av andra, mer omfattande studier, exempelvis av organisationen Gartner (2011). Men med tanke på att studien omfattar fyra företag, där två dessutom är bundna till en leverantör av proprietär mjukvara och alla arbetar med tjänster kopplade till leverantörer av proprietär mjukvara är det i efterhand ett rimligt resultat.

Sammanfattningsvis har alla företagen resonerat kring användandet av open source, de känner till dess för och nackdelar och av de sex faktorerna är det tydligt att de kopplade till vidare utveckling av mjukvaran är oviktiga för dessa företag. Ekonomin och säkerhet är däremot viktigare. Ändå använder inte dessa företag open source i deras dagliga verksamhet som en standardprogramvara. Vi drar slutsatsen att det beror på de deltagande företagens storlek och inriktning mot Microsofts tjänster. Vi tror även att om undersökningen genomförts på stora företag inom samma bransch hade resultatet på forskningsfrågan huruvida OSS är ett alternativ eller inte sett annorlunda ut. På frågan om vilka faktorer som är avgörande tror vi att de kopplade till utvecklingskategorin, och främst faktorn "Öppen källkod" som i denna studie överlag rangordnades lägst av alla hade ansetts vara viktigare.

7. Referenser

- Amaresh, A. (2013). *10 Most Popular Open Source Software Ever!*. Tillgänglig på: <http://efytimes.com/e1/fullnews.asp?edid=113521>. [Hämtad 7 Maj 2014]
- Apache Software Foundation. (2014). *Foundation project*. Tillgänglig på: http://httpd.apache.org/ABOUT_APACHE.html. [Hämtad 7 Maj 2014]
- Capek, P. G, Frank, S. P, Gerdt, S, Shields, D. (2005). *A history of IBM's open-source involvement and strategy*. IBM Systems Journal. 44 (2), s249-257.
- Carillo, K & Okoli, C. (2008). *The open source movement: A revolution in software development*. Journal of computer information systems. 49 (2), s1-9.
- Coverity. (2014). *Coverity Scan: 2013 Open Source Report*. Tillgänglig på: <http://softwareintegrity.coverity.com/rs/coverity/images/2013-Coverity-Scan-Report.pdf>. [Hämtad 05 Maj 2014]
- Daffara, C, Gonzalez-Barahona, J. (2000). *Free Software / Open Source: Information Society Opportunities for Europe*, s2-42.
- Dedrick, JD, West, JW. (2006). *Movement Ideology vs. User Pragmatism in the Organizational Adoption of Open Source Software*. Tillgänglig på: <http://www.joelwest.org/Papers/DedrickWest2008-WP.pdf> [Hämtad 11 April 2014].
- Finley, K. (2013). *Out in the Open: How to Resurrect a Dead Open Source Project*. Tillgänglig på: <http://www.wired.com/2013/12/forked/>. [Hämtad 7 Maj 2014]
- Fitzgerald, BF. (2006). The transformation of Open Source Software. MIS Quarterly, Vol. 30 No. 3, s587-598.
- Gartner (2014). *About Gartner*. Tillgänglig på: <http://www.gartner.com/technology/about.jsp>. [Hämtad 27 April 2014].
- Gartner. (2011). *Gartner Survey Reveals More than Half of Respondents Have Adopted Open-Source Software Solutions as Part of IT Strategy*. Tillgänglig på: <http://www.gartner.com/newsroom/id/1541414>. [Hämtad 12 Maj 2014]
- Hammond, T. (2014). *Windows XP support ending, but many organizations will stick with OS*. Tillgänglig på: <http://www.techproresearch.com/article/windows-xp-support-ending-but-many-users-will-stick-with-os/>. [Hämtad 7 Maj 2014]
- IBM. (2013). *IBM Commits \$1 Billion to Fuel Linux and Open Source Innovation on Power Systems*. Tillgänglig på: <http://www-03.ibm.com/press/us/en/pressrelease/41926.wss>. [Hämtad 7 Maj 2014]
- IBM. (2011). *Linux The Era of Open Innovation*. Tillgänglig på: <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/linux/transform/>. [Hämtad 7 Maj 2014]
- Jacobsen, D. (2002). *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Lund: Studentlitteratur.

- Krishnamurthy, S. (2002). *Cave or Community?: An Empirical Examination of 100 Mature Open Source Projects*. First Monday. Tillgänglig på: <http://ssrn.com/abstract=667402> [Hämtad 02 Maj 2014]
- Leonard, A. (1998). *Let my software go!*. Tillgänglig på: <http://www.salon.com/1998/03/30/feature947788266/>. [Hämtad 7 Maj 2014]
- Lerner, J. Tirole, J. (2002). *Some Simple Economics of Open Source*. The Journal of Industrial Economics, 50: 197–234.
- Magnusson, J. & Olsson, B. (2005). *Affärssystem*. Lund: Studentlitteratur.
- McHugh, J. (1998). *For the love of Hacking*. Tillgänglig på: <http://www.forbes.com/forbes/1998/0810/6203094a.html>. [Hämtad 7 Maj 2014]
- Microsoft. (2013). *A history of Windows*. Tillgänglig på: <http://windows.microsoft.com/en-us/windows/history#T1=era6>. [Hämtad 7 Maj 2014]
- Microsoft. (2014). *Informationsblad om Windows-produkters livscykel*. Tillgänglig på: <http://windows.microsoft.com/sv-se/windows/lifecycle>. [Hämtad 7 Maj 2014]
- Mitre. (2014). *Common Vulnerabilities and Exposures – Main page*. Tillgänglig på: <http://cve.mitre.org/cve/>. [Hämtad 15 Maj 2014].
- Mockus, AM. Fielding, RTF. Herbsleb, JH. (2000). *A Case Study of Open Source Software Development: The Apache Server*. ICSE '00 Proceedings of the 22nd international conference on Software engineering, s.263-272.
- Moyle, K. (2004). *Total cost of ownership and open source software*. Australien. Government of Australia, Department of Education and Children's Services.. s2-42.
- Netcraft. (2013). *August 2013 Web Server Survey*. Tillgänglig på: <http://news.netcraft.com/archives/2013/08/09/august-2013-web-server-survey.html>. [Hämtad 7 Maj 2014]
- NetMarketShare. (2014). Tillgänglig på: <http://www.netmarketshare.com>. [Hämtad 7 Maj 2014]
- Open Source Initiative. (2014a). *GNU General Public License, version 3*. Tillgänglig på: <http://opensource.org/licenses/GPL-3.0>. [Hämtad 06 Maj 2014].
- Open Source Initiative. (2014b). *The Open Source Definition (Annotated)*. Tillgänglig på: <http://opensource.org/osd.html>. [Hämtad 11 April 2014].
- Open Source Initiative. (2012). *History of the OSI*. Tillgänglig på: <http://opensource.org/history>. [Hämtad 24 April 2014].
- Pizka, MP. (2004). *Adaptation of Large-Scale Open Source Software*. European Conference on Software Maintenance and Reengineering. Tampere, Finland, March 24 - 26, 2004. IEEE.

- Quinn, L. S, Henri Andrei, K. (2013). *Comparing Microsoft Office to Open Source Alternatives*. Tillgänglig på: <http://www.idealware.org/articles/comparing-microsoft-office-open-source-alternatives>. [Hämtad 7 Maj 2014]
- Raymond, E.S. 2001. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. 1 Edition. O'Reilly Media.
- Samuelson, PS. (2006). *IBM's Pragmatic Embrace of Open Source*. Communications of the ACM. Vol 49, Nr.10. Tillgänglig på: <http://dl.acm.org.ludwig.lub.lu.se/citation.cfm?id=1164412> [Hämtad 11 April 2014].
- Schryen, GS. Rich, ER. (2010). *Increasing software security through open source or closed source development? Empirics suggest that we have asked the wrong question*. Hawaii International Conference on System Sciences (43th: 2010: Koloa, Kauai, Hawaii. Edition. IEEE Computer Society Press.
- SourceForge. (2014a). *Download Free Open Source Software - SourceForge* . Tillgänglig på: <http://sourceforge.net/directory/license:osi/>. [Hämtad 05 Maj 2014].
- SourceForge. (2014b). *Most downloads over all time*. Tillgänglig på: <http://sourceforge.net/top>. [Hämtad 7 Maj 2014]
- Tamai, T. Torimitsu, Y. (1992). *Software Lifetime and its Evolution Process over Generations*. Tillgänglig på: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.116.5313&rep=rep1&type=pdf> [Hämtad 14 Maj 2014]
- Tripathi, NT. Tiwari, VT. (2013). *Economical Benefit of Open source Technology*. International Journal of Science, Vol 2 , Nr.8. Tillgänglig på: <http://ijsetr.org/wp-content/uploads/2013/08/IJSETR-VOL-2-ISSUE-8-1587-1591.pdf> [Hämtad 07 April 2014].
- Walker, T. (2013). *20 Most Popular Open Source Software Ever*. Tillgänglig på: <http://www.tripwiremagazine.com/2010/03/20-most-popular-open-source-software-ever-2.html>. [Hämtad 7 Maj 2014].

8. Bilagor

8.1 Bilaga 1 - Intervjumall

Namn:

Befattning:

Företagsnamn:

Vad för slags företag:

Antal anställda:

Vart är ni verksamma:

Förklaring av vår uppsats: Vår uppsats behandlar området open source mjukvaror gentemot closed source mjukvaror. Det vi vill undersöka är vad för inställning företag, av er typ, har till open source och closed source mjukvaror. Vi vill fastställa hur ni går till väga när ni tar beslut om vilka mjukvaror som skall användas, om open source mjukvara ens finns med i åtanke. Och om det gör det, vilka faktorer är det som faller avgörandet. Vi har analyserat en mängd faktorer som vi anser vara av största vikt och skall med dessa frågor undersöka om de stämmer överens med er syn på frågan.

Open source enligt vår definition är mjukvara som

- Får distribueras utan begränsningar
- Där källkoden är fullt tillgänglig
- Där det är fritt fram att editera källkoden och distribuera nya versioner.

Frågor:

- Vilken är din tolkning av begreppet "Open Source Software"?
 - Presentera vår tolkning om de inte stämmer överens.
- Får anställda hos er installera de programvaror de vill på sina datorer?
 - Eller finns det en mall med godkända programvaror?
- Använder ni några open source programvaror, och då menar vi inte av någon enskild anställd utan som en av era standardprogramvaror?
 - Om ja - se Följdfrågor 1
 - Vilken typ av mjukvara är denna open source som ni använder? (Webbläsare, OS, utvecklingsverktyg, etc?)
 - Vad är den huvudsakliga orsaken till att ni valde att använda open source?
 - Gör ni någon form av analys av programvaran, på källkoden exempelvis, innan ni implementerar en open source mjukvara?
 - Hur går denna analys till?
 - Bidrar ni något till utvecklingen av dessa open source mjukvaror - antingen genom utveckling eller rapportering av buggar?
 - Om nej - se Följdfrågor 2
 - Har ni haft open source mjukvaror i åtanke över huvud taget?
- Om inte:
 - Varför fanns den typen av programvaror aldrig i åtanke?

- Vilka faktorer gjorde att ni aldrig övervägde dessa programvaror?
- Om Ja:
 - Vilka programvaror?
 - Varför fanns dessa programvaror i åtanke?
 - Vilka faktorer gjorde att valet inte föll på dessa programvaror?
- När det gäller större affärsbeslut, som exempelvis att inleda nya projekt, eller nya affärsområden etc. Hur ser den processen ut i stora drag?
 - När det gäller implementering av ny mjukvara, ser den beslutsprocessen likadan ut?
 - Hur skiljer den sig annars åt?
- När ni tar beslut om implementering av ny mjukvara, använder ni då något befintligt eller egen skapat ramverk, (beslutsmall/dokumenterat beslutsunderlag) eller dokumenterad process som stöd?
 - Om Ja:
 - Hur ser detta tillvägagångssätt ut i stora drag?
 - Om Nej:
 - Hur ser ert tillvägagångssätt ut då?
- Har ni någon rutin eller tillvägagångssätt för att se över behovet av ny mjukvara?
 - Om Ja:
 - Hur ser detta tillvägagångssätt ut i stora drag?
 - Om Nej:
 - Hur tillgodoser ni då ert behov av ny mjukvara?
- Tror du generellt sett att open source mjukvaror innehåller fler eller färre buggar och fel än vad motsvarande closed source mjukvara gör?
- Ranka vilka faktorer som är viktigast för er vid införandet av ny mjukvara (Ranka listan med viktigast till minst viktigast)
 - Kvalitet
 - Låg kostnad
 - Säkerhet
 - Tillgång till källkod
 - Anpassningsbar lösning (skräddarsydd)
 - Support
 - Framtidssäkerhet

8.2 Bilaga 2 - Intervju Fishbone

Företag: Fishbone

Antal anställda: 60 st

Verksam i: Malmö, Göteborg, Stockholm och Växjö

Typ: IT-konsult med inriktning mot Microsoft tjänster

Intervjuledare: Fredrik Källström (FK), Björn Svensson (BS)

Intervjuobjekt: Pontus Börresen (PB)

FK: *Vi börjar med att presentera lite kort våran uppsats kanske. Den behandlar området open source versus closed source, mjukvaror helt enkelt. Och det vi vill undersöka är vilken inställning ni IT företag har gentemot att använda open source inom verksamheten. Och vi vill fastställa hur ni tar beslut när ni implementerar nya mjukvaror. Och om open source är en faktor i de här besluten. Och framförallt vilka är de avgörande faktorerna i de här besluten. Det vi har gjort är att analysera ett antal faktorer, jag tror det är 8 eller 9 stycken i dagsläget som vi anser vara dem som är avgörande i mjukvaruimplementationer. Så vi kommer att med de här frågorna att förankra våra teorier i våra teorier helt enkelt. För att se om de stämmer överens. Det är tanken med intervjuerna.*

PB: *Vad är det för något ni pluggar?*

FK: *Förlåt, ja, systemvetenskap på Lunds Universitet.*

PB: *Sista terminen?*

FK: *Ja precis, sista terminen*

PB: *Spännande, vad händer då sen då?*

BS: *Sen är det, för min del blir det nog att läsa ett år till Magister.*

FK: *Och jag ska ut och jobba.*

PB: *Hur ser marknaden ut nu då?*

FK: *Den ser väl väldigt bra ut faktiskt, måste jag ju säga. Många utav oss har ju jobb långt innan vi tar studenten faktiskt.*

PB: *Ja det är så va?*

FK: *Ja så är det*

PB: *Skoj, då har man ju gjort ett rätt val*

FK: *Ja så får man ju säga*

FK: *Om vi börjar, vi brukar börja intervjuerna vad din tolkning av open source är, open source mjukvara då?*

PB: *Min tolkning av open source är ju naturligtvis olicensierad, fri för alla att både använda och framförallt bidra till. Det är väl där den stora skillnaden är. Det är så jag ser på det.*

FK: *Vi frågar för att se så att vi är på samma ungefär plan vidare inför intervjun. Vi har tagit de tre viktigaste faktorerna. Man får distribuera det utan begränsningar, källkoden är fritt tillgänglig och det är fritt fram att editera och distribuera dem nya versionerna helt enkelt. Det är dem tre viktigaste punkterna som vi ser det. Det var ju bra, det stämmer ju överens ganska bra egentligen.*

FK: *Vi fortsätter vidare med nästa fråga som är om anställda hos er får installera de programvaror de själva vill på sina datorer?*

PB: *Mm absolut*

FK: *Dem har fri tillgång?*

PB: *Fri tillgång, det finns inga som helst. Dem äger sina maskiner helt och hållet. Sen finns det ett antal centrala instanser som är mycket mer styrda. Men deras utvecklings maskiner och utvecklingsservrar gör de vad de vill med.*

FK: *Kanon, då tar vi nästa. Använder ni några open source programvaror. Och då menar jag inte enskild utan om ni använder något brett över företaget, några generella programvaror?*

PB: *Nej det gör vi faktiskt inte, mycket egentligen baserat på att vi via vårt partnerskap med... alltså vi har guld-partnerskap hos microsoft inom ett visst antal kategorier. Vilket gör att vi får använda deras produkter fritt. Och mig veterligen är det inte så att vi har någon open source faktiskt.*

FK: *Då kommer vi kanske in på följdfrågan på den egentligen, det är helt enkelt om ni har haft open source mjukvaror i åtanke över huvud taget?*

PB: *Jo men det har vi, vi utvärderar ju. Men jag tror det blir lite så per automatik att okej; kan vi ta någonting som för andra kostar då litegrann, det är nog superbra, då kör vi det för oss är det gratis. Man utvärderar nog inte alla alternativ, tror jag.*

FK: *Okej då förstår jag. Där svarade du nog nästan på följdfrågan på den också tror jag. Följdfrågorna var är vilka programvaror som är intressanta i så fall, om du känner till någonting om det?*

PB: *Vi har tittat på olika sorters källkodshantering, chattklienter innan lync blev bra, videomöten, skärmdelning. Oftast är det små små grejer.*

FK: *Små programvaror, inte de här större?*

PB: *Nej det är det inte.*

BS: *Vi kanske ska, innan ni, när ni väl blev microsoft guldparters. Innan dess hade ni något.. var det nånting..?*

PB: *Nu är det visserligen 10 år sedan, det var före min tid. Men där fanns till exempel sådana tidrapporterings-system. Den sortens saker kan jag mycket väl tänka mig att man tittade på open source. För när man startar bolaget så är ju inte det första man gör att gå*

och köpa in ett system för tidrapportering. Då tar man det som finns. Så då är jag helt övertygad om att det var mycket mer. Men nu är det väldigt mycket Microsoft, det finns ju, ju längre tiden går ju desto mer och mer har ju dem också. Dem köper ju upp företaget.

FK: *Det känns lite som att du har svarat på de två ytterliga följdfrågorna, men jag tar dem åtminstone för inspelningens skull. Vi hade också varför fanns dessa programvaror i åtanke? Men det svarade du ju på. Och så var det också vilka faktorer gjorde att valet inte föll dessa programvaror. Men det svarade du ju på att det var på grund av Microsoft.*

PB: *ja precis Microsoft.*

FK: *Ja så vi lämnar dem helt enkelt.*

FK: *Då lämnar vi lite mjukvaran för en fråga. Frågan är då när det gäller större affärsbeslut exempelvis att ta ett nytt projekt eller till och med nya affärsområdet. Hur ser den processen ut i stora drag, inte några detaljer då utan bara i stora drag?*

PB: *Man kan säga att fram tills nu så har vi varit ett helt nischat bolag inom Microsoft världen och framförallt med sharepoint som plattform. Vi har ju inte gjort något annat så då har den diskussionen nästan aldrig varit upp. Det är egentligen nu på senare tid när vi ser att, okej, sharepoint är fortfarande jättestort. Men vår kompetens hade kunnat funka inom andra plattformar, men vad skulle det kunna va. Så där tittar vi på, inte att det finns något formellt beslut att vi ska gå på någon annan plattform, det är mer nyfikenhet. Vad finns det för nånting där ut.*

FK: *Det vi tänker på är kanske mer administrativa beslut.*

PB: *Tänker du för internt bruk nu?*

FK: *Ja exakt, det kan vara att ni ska anställa någon ny eller byta kontor. Hur ser beslutsprocessen ut, finns det någon process?*

PB: *Nej det finns det egentligen inte, vi är ett litet mindre företag. Vi är ungefär 60 anställda och vi finns på 4 orter men drivs av tre regionschefer där jag ansvarar för södra sverige. Så egentligen är vi 3 regionschefer plus en VD som sitter i ledningen. Och det är svårare än att vi fattar besluten, så vi är väldigt informella där. Vi kan ta det på telefon liksom. Det finns igen process utan.. vi löser det.*

FK: *Dit vi ville komma med det, nästa fråga är helt enkelt när det gäller implementering av mjukvara om beslutsprocessen för det när man skall implementera ny mjukvara om den ser likadan ut. Om det följer samma process.*

PB: *Där skulle jag vilja säga att det kanske finns en lite tydligare process. Det finns på varje kontor och även varje central person, dem är tekniskt ansvariga. Så även om vi ska införa, om det så är ett Microsoft system eller vad det kan vara, så tas besluten hos dem. För vi fyra som driver det här bolaget är inte särskilt tekniska. Det är inte våra styrkor, vi kan aldrig sitta och bestämma vad det är för sorts mjukvaror eller hur våra datorer skall*

vara utrustade. Det är fel av oss att göra det. Så där finns tydligare process att där finns avstämningsmöten varannan vecka basis. Vad gör vi? har vi problem? behöver vi införskaffa någonting nytt? så där skulle det mycket väll kunna vara så att om vi hittar en open source lösning som är bättre, då kör vi på det. Det finns ingen som helst, att vi bara ska köra Microsoft. Att vi enbart i dagsläget konsulter utåt för kunder som har Microsoft det är en annan femma. Men vi för internt bruk hade kunnat använda en open source lösning.

FK: *Jättebra, där kom vi in lite på nästa fråga. När ni tar beslut om implementering av ny mjukvara, om ni då använder något befintlig eller egenskapat ramverk. En tydlig process för dem besluten?*

PB: *Där finns ju som sagt då en process och lite riktlinjer för det, definitivt.*

FK: *Jättebra, har ni någon rutin eller tillvägagångssätt för att se över behovet för en ny mjukvara. För att kontinuerligt kolla om det finns ett behov?*

PB: *Där finns väl ingen rutin egentligen. Folk som jobbar här har ett stort intresse för att hela tiden ligga i framkant. Så det är snarare att vi liksom, det kommer mail som.. det här behöver vi titta på och det här är jag intresserade av, kan jag testa. Kör hårt. Och är det större grejer så tar den här lilla gruppen det. Så initiativen kommer från personalen till 100%, det finns ingen från centralt håll som sitter och tittar utan det är dem som har koll.*

FK: *Så dem kommer med ideer och förslag?*

PB: *Precis, så det är deras engagemang och deras driv som ligger till grund.*

FK: *Då ska vi se, där svarade du på följdfrågan också så det var ju jättebra. Det var helt enkelt om ni inte har någon rutin, vad händer då? Men då svarade du ju att det var drivet från de anställda som tillgodoser det behovet.*

FK: *Nästa fråga. Tror du generellt sett att open source innehåller fler eller färre buggar än vad motsvarande closed source mjukvara gör?*

PB: *Ja, förr var det nog så. Nu är jag ytterst tveksam att det skulle vara fler buggar utan jag tror det är minst lika bra. Sen kanske, alltså vi jobbar oftast med lite större företag, Tetra pak Alfa laval, Peab, Lindab. Dem är ju mycket att de lägger fram 5 eller 10 års planer, då går dem på det säkra kortet, med Microsoft som de vet att det finns support i 10 år. De vet att det finns nya versioner som kommer. För dem klarar inte av att svänga sin verksamhet lika snabbt som, i alla fall är det min bild att en open source mjukvara kan ändra riktning lite fortare för det är vi som påverkar den. Tycker vi att vi skall bygga den på något annat så då kan vi göra det. Men jag tror inte att det finns fler buggar, förr kanske det var så. Men jag tror inte att det finns det nu.*

FK: *Jättebra, sista frågan så har vi ett antal faktorer. Vilka faktorer som är viktigast vid införandet av ny mjukvara. Så får du rangordna dem från 1 till 7.*

PB: *Utifrån mitt perspektiv?*

FK: *Ja utifrån din syn på det.*

BS: *Där ett är viktigast och sju minst viktig.*

PB: *Kvalitet är faktiskt det viktigaste. Framtidssäkerhet är också viktigt. Anpassningsbar lösning- 3. Support 4. Säkerhet -5. Tillgång till källkod, ja får vi det är det bra men det är ytters sällan. Låg kostnad -7.*

Pontus Börresen svarade enligt följande:

Kvalitet: 1

Låg Konstnad: 7

Säkerhet: 5

Tillgång till källkod: 6

Anpassningsbar lösning (skräddarsydd): 3

Support: 4

Framtidssäkerhet: 2

SLUT PÅ INSPELNING AV VIKT FÖR INTERVJU

8.3 Bilaga 3 – Intervju Force 12

Företag: Force 12

Antal anställda: 4 st (80 i koncernen)

Verksamma i: Lund

Typ: IT-konsult

Intervjuledare: Fredrik Källström (FK), Björn Svensson (BS)

Intervjuobjekt: Fredrik Ljungbeck(PB)

FK: *Uppsatsen behandlar open source mjukvara versus closed source mjukvara helt enkelt, och det vi vill undersöka är vad för inställning ni IT-företag har till open source mjukvaror. Och framförallt vill vi fastställa om ni använder open source mjukvaror, i vilken utsträckning, hur ni använder dem och om ni över huvud taget har dem i åtanke när ni implementerar nya mjukvaror. Så det vi har gjort är att analysera ett antal faktorer med hjälp av befintlig litteratur som vi anser vara av största vikt för open source implementationer och mjukvara implementationer i största allmänhet. Och så skall vi med de här intervjuerna försöka förankra era tankar om det för att se om ni har samma uppfattning som vi har. Så det är detta de här frågorna kommer att behandla.*

FL: *Var det dig jag pratade med i telefon?*

BS: *Nej det var mig*

FL: *Det är ju väldigt begränsat hur mycket vi använder, ni är säkert mycket mer insatta i ämnet än vad jag är. Men det kanske kan ge något i alla fall.*

FK: *Absolut, jag tror inte det skall vara några problem alls.*

FL: *Så ni får hjälpa mig med begreppen.*

FK: *Vi brukar faktiskt börja med det, så att vi är på samma plan. Så att vi är på samma nivå ungefär.*

FK: *Så första frågor är helt enkelt vilken din tolkning av begreppet open source är?*

FL: *Min tolkning.. Jag skulle nog förknippa det ganska mycket med om det är en källkod man kan komma åt. Och förändra på egen hand. Men det kanske är fel tolkning, men det är nog så jag tolkar det.*

FK: *Ja men det är den absolut största faktorn i open source, sen finns det en massa andra också. Men vi har kapat ner det till de tre viktigaste, som vi tolkar open source. Det vi bygger den här uppsatsen på egentligen. Och de är att mjukvaran får distribueras, utan begränsningar. Vem som helst får distribuera den. Källkoden är fullt tillgänglig precis som du sa och slutligen att det är fritt fram att editera källkoden och sprida nya versioner.*

FL: *Just det ja*

FK: *Så man får editera och även sprida dem nya versionerna. Det är så vi definierar open source. Och det kan du ju ha i åtanken då när du svarar på detta kanske.*

FK: *Men din tolkning är ju ungefär vad vi har också. Så jag tycker vi kör vidare med nästa fråga. Då är den om anställda hos får installera de programvaror de vill på sina egna datorer?*

FL: *Ja det får dem göra under förutsättning att det är lagliga program, de får inte installera olagliga program.*

FK: *Nej okej, men de har fria administratörsrättigheter på sin dator?*

FL: *Ja precis så det är inget vi kontrollerar heller. Men vi har skrivit in i personalhandboken så från företagsledningens sida är vi skyddade.*

FK: *Finns det någon mall, någon startmall på datorerna?*

FL: *Det har vi väl numera ja, vi har precis bytt leverantör. Dem datorerna som köps färdigkonfigurerade.*

FK: *Okej kanon. Då tar vi nästa fråga. Då är den om ni använder några open source programvaror. Och då menar vi inte riktigt om någon enskild anställd gör det utan om det används något brett över företaget, som en standardprogramvara hos er. Om någon av dem är open source?*

FL: *Programvara.. Vi jobbar ju med programutveckling. Jag kan säga exempel scriptbibliotek jQuery som jag uppfattar som open source. Det är väl inget verktyg kanske, men ett programbibliotek.*

FK: *Ja så ni använder utvecklings... alltså vad ska man säga.*

FL: *Dem program vi tillverkar åt kunderna, där utnyttjar vi open source.*

FK: *Jag förstår. Hur ser det ut på dem program ni använder internt på era datorer?*

FL: *Nej där kan jag inte påstå att vi har någon open source.*

FK: *Det är ju främst den biten vi tittar på. Även om det är intressant med den andra biten, när det kommer till utvecklingsbibliotek och sådär, programmeringsspråk och så. Men där fick vi göra en avgränsning. Men det är ändå intressant att du säger det och tar upp det, det är ändå intressant för uppsatsen. Men det är främst redan befintliga programvaror.*

FL: *Du täcker på typ Office paketet och så?*

FK: *Precis. Men ni använder inget open source internt då.*

FL: Hur är det med, jag tänker på Oracles virtual box. Nej det är det nog inte.

FK: Oracles grejer brukar generellt inte vara open source. Men jag kan kolla upp det.

FL: Det var den enda jag kunde tänka mig. På programbibliotek hade jag tänkte ut några som vi använder men..

BS: Kompilatorer som Eclipse och sådär använder ni inte?

FL: Nej, vi arbetar enbart i Microsoft miljö

FK: Men i era produkter använder ni då bibliotek som är open source

FL: Ja, jag gör en del mot ritningssystem, AutoCAD miljö. Och där finns lite toolkits och programbibliotek för att läsa en AutoCAD fil till exempel. Men det är ju som sagt bibliotek.

FK: Ja precis. Men det är bra för då har ni ändå någon form av open source inom verksamheten. Men jag tror jag får gå vidare på följdfrågorna för om ni inte har open source för uppsatsen skull.

FK: Då är en följdfråga om ni har haft open source mjukvaror för internt bruk i åtanke över huvud taget?

FL: Det har vi säkert någon gång genom tiderna men jag har inget exempel.

FK: Okej jag förstår. Är det så att.. du sa att ni jobbar i Microsoft miljö. Är det, vad ska man säga, jag försöker formulera om frågan eftersom du inte hade något exempel. Nästa fråga varför den typen av programvaror aldrig fanns i åtanke men det är ju svårt då eftersom du inte har något exempel kanske.

FK: Men har ni en bild att ni ska jobba med den här typen av mjukvaror som är bundna till ett stort företag som Microsoft?

FL: Ja det kan man nog säga att vi har haft som strategi. Force 12 är kanske lite speciellt som ett rent konsultbolag. Men om jag tänker på Avensia som är ett företag som jobbar med e-handel då som sagt så har vi haft väldigt mycket hjälp av dem i början. När man själv är lite kan man då ta rygg och rida på ett större varumärke som vi gjorde på Microsoft. Ta hjälp av dem och sen om vi kommer ut och skall sälja e-handels sajten till Kjell & Company som vi har byggt. Och vi kommer och säger att vi är 7 man som jobbar med e-handel och är jätteduktiga på det, och det kan vi ju vara. Men Kjell & Company tror liksom inte på det. Men är era leverantörer men vi jobbar med en plattform från Microsoft. Vi kan hjälpa er men skulle vi falla eller någonting så är det ändå en standardplattform.

FK: Okej, det skapar lite säkerhet?

FL: Ja. Där tror jag också faktiskt att det är viktigt. Dem kan vända sig till Microsoft och betala pengar och få hjälp.

FK: Jag förstår.

FL: Det har varit vår strategi. Nu har vi faktiskt blivit såpass stora inom området att vi faktiskt lämnar Microsoft lite och bygger en egen plattform.

FK: Ja okej. Då kan vi säga att vi nästan pratar om nästa följdfråga. Vilka faktorer det var som gjorde att ni aldrig övervägde dessa programvaror. Och det var väl som du sa då, kanske säkerheten i Microsoft.

FL: Jag tror att ge en trygghet hos kunderna. Visst även om man skulle jobba med en Linux plattform som Red Hat så kan det också ge en trygghet. För då är man väl närmare open source?

FK: Ja Red hat och Linux är ju verkligen open source

FL: Så det kan väl också ge en trygghet.

FK: Jättebra. Vi gör så att vi lämnar programvaror för en fråga nu. Och fråga är då när det gäller större affärsbeslut som exempelvis nya projekt eller nya affärsområden. Eller som ni då som skulle skapa en ny plattform. Hur ser den processen ut i stora drag? Och vi har haft lite förvirringar på denna frågan förut så jag ska tillägga att det är administrativa beslut vi menar helt enkelt, som rör verksamheten. Om det finns någon process eller rutin för det?

FL: Jag vet inte om vi har någon formell process så men.. Ja är det större som när vi ska bygga ny plattform så är det en fråga som måste upp i styrelsen och dras där. Och göra en affärsbedömning, och bland annat nu när vi valde att sluta fokusera på Microsoft. Det var en plattform Microsoft Commerce Server som vi valde att, ja, inte skylta med lika mycket. Och det ska ju upp i en formell process i styrelsen.

FK: Nu hade vi ju sån tur att vi pratade om mjukvaruplattform egentligen. För nästa fråga är om implementeringen av ny mjukvara följer samma process som affärsbeslutsprocessen?

FL: Det tror jag, alltså om det är precis samma process är det kanske inte. Men där har vi ett marknadsråd som skall komma med input från kundernas sida och så har vi ett teknikråd som skall komma med synpunkter från utvecklarna. Och så ska man försöka jämka ihop dem. Det är alltid spännande.

FK: Okej, vi kan nästan ta nästa fråga direkt nu när vi börjar närma oss den. När ni då tar beslut om implementering av ny mjukvara använder ni något befintligt ramverket eller något egenskapat ramverk, någon process för det. För att ta stöd?

FL: *Ja man kan nog säga att den utvecklingsprocessen är ganska SCRUM baserad. Sen är den väl ganska tilltydlig som den är på alla företag.*

FK: *Som det ofta är med SCRUM ja.*

FL: *Ja precis. Så man kan ju nästan jobba hur som helst.*

FK: *Så det är ett SCRUM ramverk ni använder då?*

FL: *Ja det får man nog säga*

FK: *Då tar vi nästa fråga som är lite lik. Har ni några rutiner eller tillvägagångssätt för att se över behovet av ny mjukvara. För att på förhand se om det finns eller kommer att finnas ett behov?*

FL: *Ja det får man nog säga att det är något som kommer från marknadssidan.*

FK: *Mm, där sker någon form av analysarbete?*

FL: *Jag kan ta ett exempel, om alla e-handelskunder vill ha sådan där Klarna fakturering så är det kanske något vi skall bygga in i vårt ramverk. Men det är en kundstyrt behov.*

FK: *Så det kommer utifrån då alltså.*

FL: *Det går nästan aldrig att få igenom något som utvecklare. Nu finns den där nya snabba C-sharp (C#) koden att implementera. Det går aldrig igenom i styrelsen.*

FK: *Jag förstår. Kunddrivet då kan man säga.*

FL: *Det är därifrån behovet kommer.*

FK: *Då tar vi nästa fråga. Den är om du generellt sett tror att open source mjukvara innehåller fler buggar och fel än vad motsvarande closed source mjukvara gör?*

FL: *Jag kan nog.. känslan är att det är färre men jag tror att det är en snabbare åtgärdstid i alla fall. Det är bara min känsla.*

FK: *Det är den vi är ute efter*

FL: *Känslan är att både färre och framförallt snabbare åtgärdstid.*

FK: *Så att när ett fel upptäcks så går det fortare från upptäckt till patch än vad det gör...*

FL: *Ja det är min känsla. Men det hade vart kul att höra undersökningn på det. Men vi ska inte påverka kanske.*

FK: *Nej precis men vi kan ta det efter sista frågan.*

FL: *Då har vi sista frågan där du får ranka ett antal faktorer. Vilka faktorer som är viktigast vi införandet av ny mjukvara helt enkelt. Där 1 är viktigast och 7 är minst viktigast.*

BS: *Så kan du skriva på detta papperet här.*

FL: *Så viktigast 1?*

BS: *Ja och minst viktigast 7*

Fredrik Ljungbeck svarade enligt följande:

Kvalitet: 3

Låg Konstnad: 6

Säkerhet: 2

Tillgång till källkod: 7

Anpassningsbar lösning (skräddarsydd): 4

Support: 5

Framtidssäkerhet: 1

FL: Så skulle jag nog svara

FK: Spännande. Kanon. Det var sista frågan.

SLUT PÅ INSPELNING AV VIKT FÖR INTERVJU

8.4 Bilaga 4 – Intervju Lunicore

Företag: Lunicore

Antal anställda: 50 tal timanställda (2 heltid, 7 halvtid)

Verksamma i: Lund

Typ: IT-konsult

Intervjuledare: Fredrik Källström (FK), Björn Svensson (BS)

Intervjuobjekt: Elias Gabrielsson (EG)

BS: Först och främst så kan vi börja berätta vad vi skriver om, och vi gör alltså en uppsats som behandlar området open source mjukvaror gentemot closed source mjukvaror. Och det vi tittar på då är IT-företags inställning till open source mjukvaror och closed source mjukvaror. Och då vill vi fastställa hur ni går tillväga när ni ska ta beslut om vilka mjukvaror som ska implementeras på företaget. Och även om då open source mjukvaror ens finns med i åtanke när ni gör dem här valen. Och vi har tagit fram olika faktorer då i litteraturen för att förankra detta.

EG: Ja men vi kör igång här.

BS: Vi brukar göra så att vi börjar intervjun med att fråga vilken din tolkning av begreppet open source mjukvara är?

EG: Ja, precis. När jag hör open source så tänker jag kanske på två saker, det första är när det finns communitys som utvecklar. Alltså det finns enskilda utvecklare som har ett intresse av vissa saker då, det är en utbytesform som möjliggör utveckling tillsammans över flera individer som exempel då Linux, vi har typ LibreOffice till exempel. Sen har vi det andra som jag ser att open source är ett sätt för företag att kunna, dels strategiskt att du släpper mjukvara i open source så kan du få den standardiserad, men också att det blir en väldigt transparent kommunikation mot sina kunder. Det är ett sätt att kunna leverera en produkt, så det är väll dem två. Okej det finns open source som är typ Android, till exempel när man pratar om ..? open source, sen så har du varianter nu som är lite mer organiska för att innovation. Så det är väll typ dem två begreppen som jag ser är open source.

FK: Där får du nog ta vår definition.

BS: Vår definition av begreppet är då att open source mjukvara är mjukvara som får distribueras utan begränsningar och där källkoden är fritt tillgänglig, och även där det är fritt fram att editera källkoden och distribuera nya versioner utav den. Det är så vi har definierat begreppet. Det är mest så att vi står på samma bana.

EG: Då är jag med på vad ni menar ja.

BS: Då går vi vidare, får anställda hos er installera dem programvaror dem vill på sina datorer? Och då tänker jag, nu är jag medveten om att ni har både egna datorer och att ni

har företagets datorer, så jag tänker på företagets datorer.

EG: *Ja, ja det får man och även om man då drar de till förlängningen när vi har utveckling så får man liksom ta eget initiativ till att använda dem programvaror som man ser krävs.*

BS: *Det finns alltså ingen mall med godkända programvaror?*

EG: *Nej vi har ju best practice men den syftar främst till att vi ska prata samma språk men vi vill inte begränsa våra utvecklare eller användare till vissa program som dem ska använda. Men vi har ändå i viss mån rekommendationer men dem är inte så fasta i sten om jag får uttrycka det så.*

BS: *Använder ni några open source mjukvaror? Och då tänker vi inte individuellt utan generellt på företaget.*

EG: *Ja.*

BS: *Som en slags standardprogramvara?*

EG: *Ja vi använder det. När vi utvecklar till andra så brukar vi använda open source och där har vi ju Heroku, kör ju RubyOnRails vilket är open source. Använder vi mycket WordPress, PHP, ja kompilatorer är ju open source det mesta också. Det är mycket som är open source av dem verktygen vi använder, i stora drag ja. Var det svar på frågan eller var den luddig?*

BS: *Jo nej men det tycker jag.*

FK: *De blir ju följdfrågor där.*

BS: *De blir ju följdfrågor så.*

EG: *Kör på så ser vi.*

BS: *Som du sa nu precis nästa fråga är vilken typ av mjukvara är denna open source som ni använder? Du svarade lite på det men då är det alltså?*

EG: *Det sista var ju utveckling, men vi har ju också även vissa, de flesta av oss väljer sina egna webbläsare och många vet jag använder av eget val kanske Safari eller Chrome men även också Firefox. Så det används ju open source även i dem delarna. Våra affärssystem eller vad man ska kalla det för är inte open source idag, men dem bygger in då direkt på open source komponenter i sin tur så det är en tolkningsfråga.*

BS: *Okej vad är den huvudsakliga orsaken att ni valde att använda open source?*

EG: *Stor del av det som vi använder är ju en kritisk punkt att ha human kapital som kan arbeta med saker, och det är oftast så att det är lättare för studenter på egen hand hittar intresse i saker som då är open source, som att det är lättare att ta till sig. Men det finns en naturlig koppling där att vi kanske väljer, men även att vår syn på att använda robusta frameworks som är testade och som är transparenta gör att IT-utveckling blir bättre överlag. För att bara sakerna att ha gemensamma standarder gör att allting blir mycket lättare för alla parter. Där open source är ett väldigt bra verktyg för det.*

FK: *Så det är helt enkelt transparensen då i öppen källkod som är.*

EG: *Precis och också för att vi ska kunna lova våra, i med att vårt bolag som sådant inte är jättestort så måste vi lova våra kunder att det vi utvecklar har en lång livslängd och genom att förankra det i någonting som open source så vet vi att det finns andra aktörer som också skulle kunna plocka upp andra projekt som vi har levererat. Utan att det skulle innefatta en ekonomisk bindning mot ett tredjepartsföretag som har sin egen lösning.*

BS: *Gör ni någon form av analys av programvaran, på källkoden exempelvis? Innan ni implementerar en open source mjukvara.*

EG: *Vi brukar ju inte göra reviews på källkoden själv för det har vi inte resurser till, däremot så vet jag att vi i fall där vi har behövt utvärdera tyngre system så har vi kollat på frameworket som helhet. Då är det snarare kanske inte att den är just open source, men att hur sker maintenance av den här mjukvaran och vad har den för managementsätt kring den. Det som är fördelen med kanske open source är ju främst att det är ganska högre transparens på hur en mjukvara följs upp snarare än mot en closed source.*

BS: *Följdfråga på det då, du tog de lite...Hur går den analysen till du pratade om?*

EG: *Vi brukar ju oftast utgå från kundens behov, så att har en kund vissa behov så får vi titta på om vi använder då RubyOnRails som jag nämnt och WordPress och PHP som våra inhouse frameworks för att kunna bygga human kapital inom de, så det är inte att vi vill byta för sakens skull. Men så det har nog ändå utgått där ifrån när vi hittat det här och titta om de möter våra kunders krav, när de gäller informationslagring och vilka lösningar det finns för att drifta det här och så vidare.*

BS: *Bidrar ni någonting till utvecklingen av dessa open source mjukvaror, antingen då genom utveckling eller rapportering av buggar?*

EG: *Jag skulle säga aktivt nej, företaget som sådant har ingen egen policy eller strategi för det. Däremot så kan jag tänka mig att individuella utvecklare av eget intresse kan göra det men det är ingenting som vi vet om.*

BS: *När det då gäller större affärsbeslut som exempelvis att inleda nya projekt eller nya affärsområden, hur ser den processen ut för er i stora drag?*

FK: *Vi kanske ska förtydliga den frågan för den har missuppfattats flera gånger under dem tidigare intervjuerna. Vi menar ju då alltså, ska man säga administrativa beslut, till exempel om ni ska ha nyanställda eller om ni behöver ha nyanställda liksom...så här om ni behöver utveckla själva verksamheten eller sådant liksom.*

EG: *Metoden för verksamhetsutveckling är det den eller?*

FK: *Ja alltså det handlar inte så mycket om verksamhetsutveckling egentligen endast utan det är mer administrativa beslut som påverkar verksamheten.*

BS: *Om ni ska ta ett beslut till exempel om de behövs anställas ny personal eller göra ett större inköp av någonting...byta lokal..*

FK: *Exakt byta lokal som ni nämnde ja*

EG: *Ja alltså vi har ju regelbundna ledningsgruppsmöten, vårt företag är ju ändå, trots att vi är uppemot 55 anställda så har vi inte så många jobbade timmar. Så om man säger kärngruppen i företaget har ganska hög kommunikation så att idag är det ganska mycket så att saker läggs in som på listor, idéer kastas ut kanske på någon chattgrupp och så bollas det fram och tillbaka och sen så klubbas egentligen initiativet då inom ett slags konsensusbeslut. Vi är ändå så pass få personer som har beslutsrätt idag. Det kan ju fortfarande vara så att vi använder verktygen, SWOT-analyser, tittar på fördelar, man räknar på det liksom i en naturlig del men det finns inget som egentligen är ett krav för att vi ska ta ett beslut idag.*

BS: *När det då gäller implementering av ny mjukvara ser den beslutsprocessen likadan ut?*

EG: *När det gäller mjukvara så brukar den oftast utgå väldigt tydligt i från vissa kravspecifikationer eller idéer som en kund har och där brukar vi då lämna fritt till respektive. Nu har vi ju konsulter som gör det så att man kan säga att den är i viss mån likadan, det är den. Till skillnad från de här större besluten kanske, så är det så att det är någonting som bollas lite mer i organisationen, att man lägger ut någonting på en chattgrupp eller så. Man tänker över tiden liksom att det är någonting som växer fram lite mer organiskt medans den andra grejen är lite mer att det är en tydlig problemställning där man måste ta beslut här och nu vilket gör att det blir kortare beslutsprocesser. Vilket också gör att, jag kan tänka mig att besluten inte är lika genomtänkta eller liksom brainstormade eller hur man ska uttrycka det.*

BS: *Där svarar du ju även på hur den då skiljer sig åt.*

EG: *Precis.*

BS: *När ni då tar beslut om implementering av ny mjukvara använder ni då något befintligt eller egenskapat ramverk eller dokumenterad process eller någon beslutsmall av något slag som stöd?*

EG: *Nej, eller inte vid beslut om att faktiskt implementera mjukvaran specifikt så att säga. Vi har ju en processgång för våra projekt som vi bedriver där mjukvara är en del av dem, men inte under den här funktionen använd tidigare mjukvara, den här funktionen använd, det finns inte.*

BS: *Ni har alltså ingen dokumenterad process eller några processer att stödja de här beslutet då med implementering av ny mjukvara?*

EG: *Nej det har vi inte.*

BS: *Hur ser tillvägagångsättet ut i annat fall?*

EG: Oftast så brukar vi utgå från att konsulterna får på egen hand hitta en process och bolla egentligen med dem som är närmsta chef i princip eller i projektgruppen. Hur just det här problemet ska lösas, vi har också en kunskapsåterföring tidigare genom att varje projekt testamenteras med korta meningar i princip och ges en tagg så man kan söka. Så om jag ska göra det här så kan jag se vad folk tidigare gjorde och typ lärdomar och misstag. Det är väll i för sig ett ramverk, man måste titta i kunskapsbanken innan man gör någonting men det är en själv som äger beslutet.

BS: Har ni någon rutin eller tillvägagångsätt för att se över behovet av ny mjukvara?

EG: Nej det har vi inte, det brukar komma ganska naturligt.

FK: Nej det svarade du ju nästan på förut då i med de här via konsulterna

EG: Ja precis. Här har vi inte någonting liksom som vi kan lösa då får vi..

FK: Nej.

BS: Då har du svarat på det genom att det är konsulterna som..

EG: ..Som tar beslut om det ja.

BS: Då går vi vidare. Tror du generellt sett att open source mjukvaror innehåller fler eller färre buggar och fel än vad motsvarande closed source mjukvara gör?

EG: Inte direkt egentligen, men det är lite beroende av vad man menar med motsvarande för att jag tror att här får man snarare titta på sättet man hanterar utvecklingen på än om den är open source eller closed source. Om det är så att man har en väldigt testdriven utveckling med tydligt definierat hur man hanterar fel, det tror jag snarare motverkar felen än om den är open source eller closed source. Däremot så det som är enklare med open source tror jag är att man har ett sådant test driven development att det är lättare att integrera olika parter och om man integrerar fler parter i testing så ser jag också det som att då bör man hitta fler fel snabbare. Men det finns ingenting som säger att bara för det är open source att det bör vara fler eller färre buggar skulle inte jag påstå.

FK: Nej

BS: Okej då är vi framme vid den sista frågan och då skulle vi vilja att du tittar på faktorerna.

FK: Du kan nog ta det på detta pappret.

EG: Ja

BS: Tittar på dem faktorerna och så ranka vilka faktorer som är viktigast för er vid införandet av ny mjukvara, och då ranka listan med viktigast till minst viktigast, där 1 är viktigast och 7 är minst viktigast.

Elias Gabrielsson svarade enligt följande:

Kvalitet: 3

Låg Konstnad: 1

Säkerhet: 5

Tillgång till källkod: 6

Anpassningsbar lösning (skräddarsydd): 4

Support: 7

Framtidssäkerhet: 2

SLUT PÅ INSPELNING AV VIKT FÖR INTERVJU

8.5 Bilaga 5 – Intervju Företag 4

Företag: Företag 4

Antal anställda: 30 tal

Verksamma i: Södra Sverige

Typ: IT-konsult

Intervjuledare: Fredrik Källström (FK), Björn Svensson (BS)

Intervjuobjekt: Informant 4 (I4)

NOTERING: Följande transkribering är anonymiserad, alla namn har bytts ut för att skydda identiteten hos såväl informant som organisation.

BS: *Vi kan ju kort bara berätta först lite om vad det är vi skriver om så här. Vi skriver ju då en C-uppsats som handlar då om open source mjukvaror gentemot closed source mjukvara. Och som jag sa tidigare vi tittar då på vilken inställning IT-företag har till open source och hur man resonerar kring valet mellan open source och closed source. Och vi tittar även då på om open source ens finns med i åtanke när man gör det här valet om man ens har det som ett val överhuvudtaget. Också då vilka faktorer som faller avgörandet till slut. Och vi har då analyserat en del faktorer, tittat på som vi tycker är viktiga i det här valet och ska se om det överensstämmer med er bild av det.*

I4: *Absolut.*

BS: *Då ska vi se. Tänkte börja på...Kort fråga, vilken tolkning du har av begreppet open source software?*

I4: *Jag skulle väl personligen säga alltså mjukvara som där källkoden till mjukvaran ligger upplagd på något publikt källkodshanteringssystem typ Github eller Codeplex eller Bitbucket eller vad det nu kan vara. Där man dels då kan bygga sina egna byggen av mjukvaran eller liksom modifiera och naturligtvis även bidra. Samtidigt som man ska kunna ladda ner och köra det. Gratis tänker jag först och främst men det är klart att det finns väl kanske open source som även inte är gratis, för enterprise utnyttjande till exempel. Men att dem ofta är gratis kanske då för personligt användande. Men min första tanke är väl oftast gratis mjukvara där källkoden ligger öppen att bidra till eller modifiera om man vill. Det skulle jag nog säga är min tolkning.*

FK: *Ja det är ju ganska precis våran tolkning också.*

I4: *Ja, ja men det tror jag, det är så jag ser på det i alla fall.*

BS: *Då står vi på samma känns det som.*

BS: *Då fortsätter vi. Får anställda hos er installera de programvaror dem vill på sina datorer?*

I4: *Ja det får dem.*

BS: *Det får dem.*

I4: *Vi har en ganska öppen inställning till mjukvara och framförallt open source också. Våra utvecklare får använda dem programvaror och mjukvaror som dem känner att dem behöver i sitt arbete som underlättar. Vi har naturligtvis några färdiga eller liksom några program som vi oftast alltid installerar just för att dem är användbara, som är både open source och closed source naturligtvis. Men vi vill inte begränsa eller förhindra någon, om Bertil tycker att det här programmet är väldigt bra i hans arbetsflöde så då får han göra*

det. Så vi vill inte hindra något. Och kan han då kanske, eller Greta har något tips på ett program som någon annan kan utnyttja så då försöker vi bara promota det så mycket som möjligt. Så om någon annan kan dra nytta av det med, så att vi försöker ha en väldigt öppen inställning kring det. Naturligtvis under ansvar, menar att ge ett 30-tal personer admin rättigheter på sina datorer inom ett nätverk där vi behandlar säkra eller liksom sekretessbelagda data, det kräver ju att man arbetar under ett visst ansvar och tänker på vad man installerar och vad man delar med sig av i mail och andra uppdrag. Vi får ju naturligtvis virus då och då men den biten hanterar vi centralt med automatiska virussökningar och sådant, så att det är väll egentligen det som är den största risken egentligen med att låta folk installera vad dem vill. Men även om dem flesta är väldigt erfarna datoranvändare så händer det ju att man kommer in på fel sida ibland. Så att korta svaret är Ja.

BS: *Ja det är bra.*

BS: *Använder ni några open source programvaror? Och då tänker jag inte som på enskilda användare utan som en standardprogramvara.*

I4: *Ja, det tror jag i alla fall, jag blev lite osäker nu för jag funderade på den här innan, och alltså vi eller jag använder framförallt då, nästan alla använder någon form av lite mer avancerade textredigerare. Sublime text använder vi och jag kommer inte ihåg om den är open source eller inte, jag tror nästan det.*

FK: *Jag tror också det, men om inte så ska jag kolla upp det i efterhand i alla fall.*

I4: *Och Notepad++, det är dem två jag kommer på direkt. Sen vet jag inte hur det är med Paint.net, det kan va att det är, jag ska bara kolla här, det kanske inte är det.*

FK: *Det är Freeware i alla fall.*

I4: *Precis.*

FK: *Alla dem här tre.*

I4: *Precis det är nog därför jag liksom associerar dem med open source därför att de ofta är gratis. Funderar på om det är några andra här som vi sitter med lokalt. Webbläsare och så då kör vi nästan alla som finns i alla fall, Firefox, IE, Chrome. Men det är ingen egentligen där som är open source, inte ens Chrome tror jag. Eller i för sig den drivs av webbkit och det är väll...*

FK: *Mozilla är ju lite semi open source kan man säga.*

I4: *Så att det är så fall om man klassar det som lite halv open source så fall. Sen är det ju så att, vi är ju ett företag som är specialiserade på Microsoft produkter, och TeknikX då framför allt så att av naturen blir det ju mycket Microsoft och där är det ju tyvärr inte så mycket som är open source. Så att dem flesta utvecklingsverktygen och serverversioner och sådant som vi använder är ju Microsoft produkter. Därav har vi kanske inte så pass mycket open source när det kommer till den biten, men just dem här verktygen som vi använder som en hjälp eller tillägg till utvecklingsverktyg och sådant, där har vi en hel del halvöppna verktyg skulle jag tro. Men det finns ju mycket sådana här tillägg till Visual Studio också som säkert är open source. Jag kommer inte på några direkta just nu, jag kan kolla det under tiden här...*

BS: *Vad är den huvudsakliga orsaken till att ni valde att använda open source på dem här bitarna då, om vi utgår från dem som vi pratar om här?*

I4: *Det handlar nog dels om att de är dem, i alla fall de som jag nämnde där nu Sublime, Notepad++ till exempel om det nu är open source, jag blir lite osäker här nu. Så handlar det så fall om att de är bra verktyg som många använder, det finns liksom dem mest stabila. Framför allt också att dem är gratis och använda, så att det är snabbt att bara ladda ner och bredda. Börja använda i arbetet när man vill, behöver inte liksom kosta på dem här licenser eller något sådant. Om man bara behöver göra något snabbt. Sen naturligtvis vissa program så ska man kanske skaffa licenser till just för lite större användning, men framför allt just bara för att dem är bra liksom, bra verktyg, dem underlättar väldigt mycket. Så att det är väl det främsta.*

BS: *Gör ni någon analys av källkoden.*

I4: *Nej.*

BS: *Innan ni implementerar den?*

I4: *Nej, inte något direkt. Som sagt nu använder vi kanske mest programvara där som använts sen innan, så att då skulle det kanske vara något helt otestat så kanske vi skulle kolla igenom lite. Jag vet jag använt något verktyg ibland, eller bara det är någon enskild gång som för migrering av en lösning till en annan, som va open source. Och då var jag tvungen att modifiera det verktyget i koden för att det skulle passa den här nya lösningen som vi skulle migrera det till, och visst då blev det ju en analys av koden. Men nej inget formellt.*

BS: *Nej.*

BS: *Bidrar ni någonting till utvecklingen av dessa open source programvaror? Antingen utveckling eller bugg-rapportering kanske.*

I4: *Tyvärr inte. Inga sådana liksom distanser, vi har haft lite projekt där vi har tänkt att...eller projekt, lite sådana här Hackathons och sådana. Tänkt att nu ska vi göra något som vi kan få upp på GitHub, som vi kan få lite respons på. Lite bra feedback på men tyvärr har det inte blivit något konkret där än, men vi har planer på det. Men det är så fall bara om det är någon som sitter privat och hjälper till men inte liksom något uttalat från företaget.*

FK: *Nej, förlåt får jag komma med en följdfråga på det. Du sa det här när du använder en open source för migrering. La du upp den förändrade, du spred inte den?*

I4: *Nej, det gjorde jag inte, just för att det här var ett så specifikt fall för den kunden.*

FK: *Ja okej jag förstår.*

I4: *Det är möjligt att någon hade gynnat av det men min tanke då var att det här är, jag fick skriva om det just för att göra en specifik migrering av särskild data för bara den kunden så att jag tror inte att det hade gynnat.*

FK: *Nej nej jag förstår.*

I4: *Men absolut det hade man naturligtvis kunnat göra.*

BS: *När det gäller större affärsbeslut, exempelvis att inleda nya projekt eller nya affärsområden. Hur ser den processen ut i stora drag för er?*

I4: *Vi har ju en arbetsprocess som vi kallar för ProcessX, och det är ju en variant av, eller vi inkorporerar lite agila tekniker, scrum och MomentX då. Egentligen handlar det om hela vår utvecklingsprocess. Men jag vet inte om det är den du syftar på eller tänker du på själva anförskaffandet av programvaran eller?*

FK: *Det kommer vi nog till.*

BS: *Ja vi kommer till det.*

FK: *Nu är det mer såhär beslut som kanske rör verksamheten egentligen och vart den ska ta sig egentligen.*

I4: *Det är nog inte jätte formellt där heller, men ofta går det till så att är det någon som har ett program eller något verktyg där man använder när dem utvecklar som då kanske testats privat eller bara testats personligt i något projekt. Så visar det sig att det funkar bra, är det gratis så naturligtvis så kan vi försöka få ut det till alla andra också, igenom egentligen bara säga att det här är ett bra verktyg. Ibland så förinstallerar vi det då på utvecklingsmaskiner. Är det någonting som då behöver köpas licenser till så går det kanske inte, men just alltså då går jag till min...Jag är ju teknikchef då, så då går jag till chefen som är ansvarig för pengar.*

BS: *Hehe, ja precis.*

I4: *Och ber där, om vi kan köpa in licenser till det här, hur många vi i så fall behöver och sen så distribuerar vi det egentligen bara via, genom att vi ger ett visst antal då som kanske har behov av den här mjukvaran då licenser till, så lagrar vi licenserna naturligtvis då säkert i våra system. Men vi har ingen sådan super formell process kring det heller utan när man är i behov så tar vi upp det liksom.*

BS: *Men skulle du säga då att processen för de här med implementering av mjukvara är likadan som beslutsprocessen för andra verksamhetsbeslut som inte har med implementering av mjukvara att göra?*

I4: *Nja det skiljer sig nog lite beroende på vad det är, vad det gäller liksom, ska vi köpa in nya datorer till exempel då är det en lite större investering oftast. När det gäller utvecklingsverktyg och sådant så dels så har vi ett samarbete eller partnerskap med Microsoft som gör att all mjukvara som Microsoft då, där får vi lite bättre erbjudanden och vi får sådan här MSDN licenser då naturligtvis via dem så att vi kan utnyttja mjukvaran på ett enklare sätt och behöver inte göra någon stor process av det. Men som sagt ska vi köpa in nya datorer till exempel så kostar det ju lite mer och det är ju en engångsinvestering så att då tar man det när det ska göras. Men nja, jag har inte varit i den positionen som jag är i just nu så pass länge heller så att jag kan inte säga om vi haft något annat tillvägagångssätt före innan heller.*

FK: *Ja men det är jättebra.*

BS: *När ni tar beslut om implementering av ny mjukvara, använder ni då något befintligt eller egenskapat ramverk eller en beslutsmall eller dokumenterad process?*

I4: *Nja alltså oftast om det är program som vi återanvänder i det vanliga bruket, då har vi ett system eller ett program som heter System Center Management som Microsoft produkt också, som gör att vi kan trycka ut färdig installation av Windows med den, dem programvarorna vi vill göra tillgängliga. Så att det är bara att vi trycker in ett USB-minne och så installerar den på datorn och så drar den ner allting som behövs. Där har vi egentligen bara satt liksom basprogrammen som vi behöver, typ Office och ja kanske något tidrapporteringssystem och sådant där som vi tar med. Men sen är det som sagt, sen är det upp till var och en att installera egna grejer om man har behov och vill ha det. När det gäller utveckling sen då har vi färdiga virtuella maskiner som där det då är färdiginstallerade utvecklingsverktyg på, men dem är också naturligtvis fria att ändra på och modifiera och vill man lägga till något så är det upp till var och en. Så att vi har inte superdokumenterat där heller men det finns en slags process, det gör det.*

BS: *Har ni någon rutin eller tillvägagångssätt för att se över behovet av ny mjukvara?*

I4: *Nej egentligen inget konkret där heller, som sagt är det någon som har behov av det så. Vi är ett ganska litet företag så är de någon som, då är det bara att skicka ett mail till mig. Eller är det någon som behöver någonting så kan man säga till liksom. Och ofta, det blir lite såhär också att vi sitter i ganska tigha team, om det är ett bra program liksom då sprider det sig runt om i teamet, sen kanske även då utåt. Vi försöker ofta ha sådana här vad vi kallar teknikkvällar där vi kanske berättar om det är någon person eller några personer som berättar om någon ny häftig teknik, eller det kan vara ett nytt program eller bara ett projekt som dem arbetat på. Då kan det ofta komma fram sådana saker då, vi kanske då tar i tu med och lägger in så det blir ett standardprogram för alla.*

FK: *De är ju även om det inte är rutin så är det ändå någon form av spridande event kan man ju säga då...*

I4: *Ja men precis det kan man säga...men ja, det sker ganska organiskt liksom. Men det handlar om att vi fortfarande också är ganska små, relativt sett. Även om vi nu då är en del av Företag 5 också som är jättestora, men vi är fortfarande vår enhet här och vi sköter vårt på det sätt vi tycker funkar bäst. Då blir det mycket att man behöver inte ha några formella processer för allting. Och inser vi sen att det, för det händer ju att vi ser att det här är någonting som många använder, här är något problem som folk löser på samma sätt, då är det ju så att då implementerar vi det så att det följer med i standardinstallationen av Windows till var och ens dator eller i färdiga mallar för våra utvecklingsmiljöer.*

BS: *Tror du generellt sett att open source mjukvaror innehåller fler eller färre buggar än motsvarande closed source mjukvara gör?*

I4: *Jag tror det är fifty-fifty där alltså. Jag har använt många program som varit closed source som har skitjobbiga buggar och det finns ju open source som har farliga buggar också, typ HeartBleed. Jag tror inte den ena eller den andra behöver vara bättre egentligen. Det finns ju fördelar med open source som att det finns många som kanske bidrar och då kan hitta buggar och när de hittas buggar kanske de fixas snabbare. Kanske, det skiljer sig också naturligtvis men dem kanske inte har en lika strikt testningsprocess som vissa closed source programvaror men å andra sidan allting beror ju på vilket det är och vad det är och vem som utvecklar det liksom. Nej jag skulle inte vilja säga att det ena är bättre än det andra utan det är nog som sagt fifty-fifty det där. Men det är min personliga åsikt.*

BS: *Så den sista frågan, då skulle vi vilja be dig att om du skulle vilja ranka dessa faktorerna i listan? Vilka faktorer som är viktigast för er vid införandet av ny mjukvara, och då ranka listan viktigast till minst viktigast där alla faktorer blir rankade från 1-7.*

FK: *Du kan nästan ta de på det här pappret kanske, det är det vi sparar.*

BS: *Du kan bara skriva siffrorna.*

I4: *Jag börjar med 1 vad som är viktigast?*

BS: *Ja precis och 7 är minst viktigast.*

I4: *Allright. Nej men alltså spontant så känns väll kvalité som...är det någonting som är bra så som vi har nytta av så är det ju det som talar för att vi ska använda det, menar vi använder verktyg som är closed source men som ger oss mycket fördelar i utveckling, som kostar naturligtvis också. Men är det bra så är det bra och då börjar vi använda det. Men*

ja, det skulle väll va låg kostnad kanske 2 då. Framtidssäkerhet kanske är 3, säkerhet 4. Kvalité först, låg kostnad 2, framtidssäkerhet 3, det är egentligen såhär jag ser det i alla fall. Det är det jag ser som gör det lätt och snabbt att avgöra om det är någonting vi kan ha nytta av. Är det en bra produkt, är det låg kostnad så att vi kanske kan installera det gratis eller om vi talar om en liten licenskostnad så talar ju det för. Framtidssäkert, alltså det går ju väldigt snabbt när vi i IT-världen, så att det är klart att är det någonting som vi ser kan funka framöver så talar ju det för det också. Jag kryssade för säkerhet för att det kan ju vara känslig data vi jobbar med och då är det bra om man kan förlita sig på produkten. Support, det är alltid bra om det finns någon community eller dokumentation kring mjukvaran så att det i fall det är något problem som uppstår eller om man kan använda det på något sätt som man inte har tänkt innan. Anpassningsbar lösning, jag ser inte det som något jättestort krav, samma sak egentligen tillgång till källkod, eftersom vi så sällan gör analyser av källkoden där vi faktiskt går in och tittar i den utan det är nog bara ett plus liksom.

Informant 4 svarade enligt följande:

Kvalitet: 1

Låg Kostnad: 2

Säkerhet: 4

Tillgång till källkod: 7

Anpassningsbar lösning (skräddarsydd): 6

Support: 5

Framtidssäkerhet: 3

SLUT PÅ INSPELNING AV VIKT FÖR INTERVJU