

Mjukvarukvalitet i öppen programvara

- en explorativ och kvalitativ studie av uppfattningar om mjukvarukvalitet

Kandidatuppsats, 15 högskolepoäng, INFK01 i informatik

Framlagd: 2008-01-22

Författare: Jakob Mattelin
Jonas Persson

Handledare: Claus Persson

Examinatorer: Kjell-Åke Holmberg, Hans Lundin

Mjukvarukvalitet i öppenprogramvara

- en explorativ kvalitativ studie av uppfattningar om mjukvarukvalitet

© Jakob Mattelin och Jonas Persson

Abstract

Open source software has recently been discussed, not least in media as a revolution in IT. A number of advantages have been listed in changing proprietary software into open source software. Examples of such pros are no expenses for license fees and improved efficiency. In spite of this, few users have made this change. One of the reasons is a distrust of the quality in open source software. We have made an investigation how the quality in the open source software is estimated among persons who are responsible, or have a good insight in decisions regarding the choice of software in cooperation's or organizations. With ISO/IEC 9126's definition in view, regarding quality of open source software, four semi-structured interviews were undertaken with subjects mentioned above. Between respondents a great variability of opinions were found. It was shown that some of their opinions differ in several aspects from scientific surveys regarding the quality of open source software. This led us to the conclusion that the distrust of the quality is isolated to a couple of specific attributes. Furthermore we concluded that some attributes are perceived as having low quality although scientific surveys points in another direction.

Keywords: Open source software, software quality, ISO/IEC 9126, FLOSS

Kandidatuppsats framlagd: 2008-01-22

Omfång: 103 sidor

Handledare: Claus Persson

Förord

Denna rapport avser att vara ett examensarbete på det Systemvetenskapliga programmet vid Institutionen för Informatik, Lunds Universitet. Rapporten representerar förhoppningsvis också slutskedet av våra studier då jobb väntar oss båda efter examination. Vi vill därför passa på att tacka hela institutionen och dess lärare för en utbildning som gett oss goda kunskaper för att förhoppningsvis kunna axla de arbeten som redan väntar oss.

Studien hade inte varit möjlig att utföra utan de respondenter som ställt upp. Vi är väldigt tacksamma för att ni tagit er all tid för att hjälpa oss vid insamlingen av data. Vi vill även tacka adjunkt Claus Persson och övriga deltagare i vår handledningsgrupp som varit till nytta med handledning och goda råd.

En ambition med rapporten har varit att den skall ha en god ”begriplighet”. Vi vill därför tacka docent Bernt Falk som uppger sig representera ”den icke IT kunnige” lekmannen. Han har visat ett stort engagemang och gett stilistiska råd som därmed gjort rapporten mer ”användarvänlig”.

Tack!

Jonas Persson och Jakob Mattelin

Studerande vid Institutionen för Informatik, Lunds Universitet 2008-01-15

Innehållsförteckning

1. BAKGRUND	1
1.1 ÖPPEN PROGRAMVARAS GENOMBROTT DRÖJER TROTS FLERTALET FÖRDELAR	1
1.2 PROBLEMFÖRMULERING	2
1.3 SYFTE.....	2
1.4 FRÅGESTÄLLNINGAR	2
1.5 AVGRÄNSNINGAR.....	3
1.6 BEGREPPSDEFINITIONER	4
1.6.1 <i>Fri mjukvara och öppen källkod</i>	4
1.6.2 <i>Proprietär programvara</i>	4
2. METOD	5
2.1 VETENSKAPSTEORETISK STÅNDPUNKT	5
2.1.1 <i>Vetenskaplig kunskapsuppfattning</i>	5
2.1.2 <i>Kvantitativ och kvalitativ forskningsmetod</i>	6
2.1.3 <i>Forskningsansats</i>	6
2.2 VALIDITET OCH RELIABILITET	6
2.3 ETIK	7
2.4 GENOMFÖRANDET	8
2.4.1 <i>Litteratursökning och kritisk granskning</i>	8
2.4.2 <i>Urval av respondenter</i>	9
2.4.3 <i>Intervjuer</i>	10
2.4.4 <i>Transkriberingar</i>	11
2.4.5 <i>Analys av intervjuer</i>	12
2.5 SPRÅKLIGA RESERVATIONER	13
3. TEORETISK REFERENSRAM	14
3.1 PRODUKT- OCH MJUKVARUKVALITET	14
3.1.1 <i>Respondenternas kvalitetsperspektiv</i>	15
3.1.2 <i>Mjukvarukvalitetsmodeller och dess överrensstämmelse med kvalitetsperspektiv</i>	16
3.2 KVALITETSATTRIBUT.....	17
3.2.1 <i>Funktionalitet</i>	17
3.2.2 <i>Tillförlitlighet</i>	17
3.2.3 <i>Användbarhet</i>	18
3.2.4 <i>Produktivitet</i>	18
3.2.5 <i>Underhållsmässighet</i>	18
3.2.6 <i>Flyttbarhet</i>	19
3.3 FORSKNINGENS SYN PÅ MJUKVARUKVALITET I ÖPPEN PROGRAMVARA	19
3.3.1 <i>Forskningens syn på funktionalitet</i>	20
3.3.2 <i>Forskningens syn på tillförlitlighet</i>	21
3.3.3 <i>Forskningens syn på användbarhet</i>	21
3.3.4 <i>Forskningens syn på produktivitet</i>	21
3.3.5 <i>Forskningens syn på underhållsmässighet</i>	22
3.3.6 <i>Forskningens syn på flyttbarhet</i>	23
4. UNDERSÖKNINGEN	24
4.1 RESPONDENTER, FÖRETAG OCH ORGANISATIONER	24
4.2 EMPIRISKT RESULTAT	25
4.2.1 <i>Respondenternas uppfattningar om funktionalitet</i>	25

4.2.2 Respondenternas uppfattningar om tillförlitlighet	26
4.2.3 Respondenternas uppfattningar om användbarhet.....	27
4.2.4 Respondenternas uppfattningar om produktivitet	28
4.2.5 Respondenternas uppfattningar om underhållsmässighet	28
4.2.6 Respondenternas uppfattningar om flyttbarhet.....	30
5. ANALYS - UPPFATTNINGAR I RELATION TILL FORSKNING	31
5.1 FUNKTIONALITET I ÖPPEN PROGRAMVARA	31
5.2 TILLFÖRLITLIGHET I ÖPPEN PROGRAMVARA	32
5.3 ANVÄNDBARHET I ÖPPEN PROGRAMVARA	33
5.4 PRODUKTIVITET I ÖPPEN PROGRAMVARA.....	34
5.5 UNDERHÅLLSMÄSSIGHET I ÖPPEN PROGRAMVARA	34
5.6 FLYTTBARHET I ÖPPEN PROGRAMVARA	35
6. SLUTSATSER OCH DISKUSSION.....	37
6.1 FRÅGESTÄLLNING ETT.....	37
6.2 FRÅGESTÄLLNING TVÅ	38
6.3 EGNA REFLEKTIONER.....	40
6.4 FÖRSLAG TILL FORTSATT FORSKNING	41
7. REFERENSER.....	42
8. BILAGOR	44
8.1 FÖRETAGSBREV	44
8.2 RESPONDENTVALIDERINGS BREV	45
8.3 INTERVJUGUIDE	46
8.4 TRANSKRIPTIONSBILAGOR	48
8.4.1 Bank.....	48
8.4.2 Produktutvecklingsföretag	66
8.4.3 Vårdenhet.....	77
8.4.4 Skola.....	90

Tabellförteckning

TABELL 2-1 – SÖKORD OCH DATABASES	8
TABELL 2-2 - KODER OCH TEMAN	13
TABELL 5-1 – ”FUNKTIONALITET” I ÖPPEN PROGRAMVARA	32
TABELL 5-2 – ”TILLFÖRLITLIGHET” I ÖPPEN PROGRAMVARA	32
TABELL 5-3 – ”ANVÄNDBARHET” I ÖPPEN PROGRAMVARA	33
TABELL 5-4 – ”PRODUKTIVITET” I ÖPPEN PROGRAMVARA	34
TABELL 5-5 – ”UNDERHÅLLSMÄSSIGHET” I ÖPPEN PROGRAMVARA	35
TABELL 5-6 – ”FLYTTBARHET” I ÖPPEN PROGRAMVARA	36

1. Bakgrund

1.1 Öppen programvaras genombrott dröjer trots flertalet fördelar

På senaste tiden har öppen programvara (se kap. 1.6.1) beskrivits av allt fler som en revolution inom IT-industrin. Flertalet fördelar framföres för att gå över till öppen programvara. Ett par av dessa är: effektivisering av programvara, ökad kostnadseffektivitet samt att system skall kunna utvecklas bättre med öppen källkod. Trots detta, har vi inte sett något genombrott för öppen programvara. (Ny Teknik 2007)

Statskontoret (2003) har i en rapport belyst de ekonomiska fördelarna med öppen programvara. Exempelvis sänkta kostnader och förenklad licenshantering. Sveriges Radio (2003) framhåller ekonomiska förtjänster med användning av öppen programvara. De skriver att den svenska staten skulle kunna spara in miljardbelopp på att byta ut existerande proprietär programvara (se kap. 1.6.2) mot öppen programvara. Ovanstående exempel är bara ett axplock ur studier som visat på ekonomiska förtjänster med öppen programvara, vi anser därmed detta påstående vara väldokumenterat.

Att minska utgifter, för att maximera vinst i en verksamhet, är ett allmänt känt tillvägagångssätt och således kan man fråga sig varför inte fler följer denna linje. Det borde alltså finnas andra faktorer som påverkat besluten kring val av programvara.

Côte, Suryn och Georgiadou (2007) skriver att under det senaste årtiondet har en större del av mjukvaruindustrin skiftat fokus från att utöka funktionaliteten i programvara till att istället sträva efter att förbättra vad de kallar ”*user experience*”. Kort sammanfattat, inbegriper dessa förbättringar av egenskaper i slutprodukten, som ökning av stabilitet, pålitlighet, säkerhet och enkelhet att använda. Detta leder till, att slutanvändaren upplever högre grad av mjukvarukvalitet vid användning av programvara.

Andra som uppmärksammat betydelsen av mjukvarukvalitet är Laplante, Gold och Costello (2007). De hävdar att det finns en utbredd misstro mot att öppen programvara också skulle uppvisa god mjukvarukvalitet. Detta medför, att väldigt få företag och organisationer använder sig av öppen programvara. Misstron saknar dock grund i forskningen. Istället hävdar Laplante et al (2007) att öppen programvara kan uppnå högre mjukvarukvalitet än sluten programvara.

1.2 Problemformulering

Laplante et al (2007) skriver om en allmän skepsis mot mjukvarukvalitén i öppen programvara. Vilka attribut av mjukvarukvalitet som denna skepsis grundar sig på är dock oklart. En förståelse för, hur nuvarande och blivande användare, uppfattar mjukvarukvalitén i öppen programvara är nödvändig. Detta för att utvecklare av öppen programvara skall kunna förbättra aspekter av programvaran som uppfattas som bristfälliga. Ökad mjukvarukvalitet i programvaran kan då leda till större användning i kommersiella syften och ”revolutionen” kan börja.

Riktigt så enkelt är det dock inte. Laplante et al (2007) hävdar att misstron mot mjukvarukvalitet i öppen programvara inte har vetenskapliga belägg. Istället menar Laplante et al, att öppen programvara kan uppnå högre mjukvarukvalitet än slutna programvara. Forskning som visar på hög mjukvarukvalitet i öppen programvara förefaller ologisk, med tanke på den skepsis som ovan nämnts.

Uppfattningar om mjukvarukvalitet i öppen programvara, bland personer som beslutar om val av programvara i företag, skiljer sig uppenbarligen från forskning i detta ämne. Tawileh & Rana (2006) hävdar att mjukvarukvalitén i öppen programvara är hög, men utvecklarna misslyckas med att nå ut med detta budskap till användarna. De skriver dock inte om det är någon specifik aspekt av mjukvarukvalitet som inte förmedlats korrekt till användarna. Förståelse för detta, innebär att utvecklarna kan utöka arbetet med att nå ut med information till marknaden avseende dessa aspekter.

1.3 Syfte

Syftet med uppsatsen är, att beskriva hur öppen programvaras mjukvarukvalitet, enligt ISO/IEC 9126:1991, uppfattas av personer som beslutar om val av programvara i företag och organisationer. Dessutom avser uppsatsen redogöra för, om dessa uppfattningar stämmer överens med vad forskningen anser om mjukvarukvalitet i öppen programvara.

1.4 Frågeställningar

Uppsatsens syfte kommer att uppfyllas genom att följande två frågeställningar besvaras:

Frågeställning 1: Hur uppfattas mjukvarukvalitet i öppen programvara av personer som beslutar om val av programvara i företag och organisationer?

Frågeställning 2: Vad finns det för likheter och skillnader mellan dessa uppfattningar jämfört med vad forskningen anser om mjukvarukvalitet i öppen programvara?

1.5 Avgränsningar

Vi kommer i denna uppsats endast att bidra till en förståelse för hur öppen programvara uppfattas med avseende på mjukvarukvalitet. Vi kommer inte att förklara hur dessa uppfattningar bildats. Inte heller kommer vår uppsats att visa på trender över tid vad gäller dessa uppfattningar, då vår undersökning endast är en engångsundersökning av hur det ser ut i dagsläget. Vi kommer inte heller att empiriskt undersöka hur enskilda program upplevs av våra respondenter. Istället avser vi att bidra med en generell bild av uppfattningar om öppen programvara.

De uppfattningar som redovisas i studien är uttryckta av professionella IT-personer och speglar deras yrkesmässiga överväganden. Fokus för undersökningen riktas därmed på beslutsfattarnas egna uppfattningar om öppen programvaras mjukvarukvalitet. Uppsatsen presenteras som en explorativ studie vilket huvudsakligen innebär att vi endast har ambitionen att bidra till skapande av underlag och uppslag till framtida forskning. Detta innebär också att vi inte har för avsikt att beskriva samtliga uppfattningar om mjukvarukvalitet inom samtliga branscher. Studien är gjord för att undersöka om det finns någon skillnad mellan studiens respondenter och forskningen.

Uppsatsen kommer inte att redogöra för vilka attribut av mjukvarukvalitet som respondenterna anser vara viktiga för den egna organisationen. Således kan ej undersökningens resultat ligga till grund för förståelse till varför öppen programvara inte används i större utsträckning. Det kan exempelvis vara så att ett antal attribut av mjukvarukvalitet upplevs vara av mycket låg kvalitet. Om de saknar betydelse för organisationen går det dock ej att hävda att de är orsak till det låga användandet av öppen programvara i företag och organisationer.

Om det visar sig att någon av våra respondenter utvecklar programvara, så kommer vi ej granska uppfattningar om de egenutvecklade programvarorna. Detta tror vi hade inneburit en risk för bias då de troligtvis vill framföra sin egen programvara i god dager.

1.6 Begreppsdefinitioner

1.6.1 Fri mjukvara och öppen källkod

Öppen källkod är ett begrepp som myntades av Eric S Raymond (1997, enligt Computer Sweden 2007) i ”The Cathedral and the Bazaar”. Samtidigt så existerade en liknande parallell ideologi med begreppet ”fri mjukvara”. De stora skillnaderna mellan de båda begreppen är att fri mjukvara syftar på att göra koden fri för exempelvis modifiering av användaren, medan öppen källkod pekar på hur mycket bättre programmen blir om många hjälper till med utvecklingen (Computer Sweden, 2007). På grund av dessa ideologiska skillnader, och de förvirringar de ledde till, så samlades de båda definitionerna i ett nytt begrepp, FOSS (Free and Open Source Software), dock utan att ta parti för någon av ideologierna. Akronymen FOSS har senare ändrats till FLOSS (Free/Libre Open Source Software) eftersom definitionen av "Free" ansågs kunna bli missstolkad då "Free" endast syftar på "fri" mjukvara och inte gratis. Den svenska översättningen av FLOSS är "öppen programvara" och definieras av Statskontoret (2003) som en programvara användaren fritt får använda, kopiera, distribuera, undersöka, ändra och förbättra den.

1.6.2 Proprietär programvara

Alternativet till öppen programvara är enligt Statskontoret (2003) den proprietära programvaran. En proprietär programvara tillhandahålls med strikta villkor för hur den får användas. Den får bland annat inte vidareutvecklas eller distribueras. Statskontoret (2003) skriver att Microsoft med sina proprietära produkter har en marknadsandel på 85-90 % inom ett antal produktområden. Detta visar att proprietär programvara inom flera områden, i dagsläget, är klart mest använd. Begreppet sluten programvara används som synonym till proprietär programvara, framförallt av våra respondenter.

2. Metod

I detta kapitel ämnar vi beskriva hur vi gått tillväga vid genomförandet av studien. Detta görs för att underlätta för replikation och evaluering av metoden enligt Backman (1998). För att kunna möjliggöra en replikation presenterar vi bland annat en genomförandefas i detta kapitel. På så vis är det möjligt för övriga att upprepa metoden under motsvarande förhållanden. Detta innebär, att det blir möjligt att kontrollera de resultat som presenteras i uppsatsen.

Evalueringens innebörd är enligt Backman, att utomstående skall kunna värdera eller bedöma det empiriska tillvägagångssättet. Vi syftar således till att motivera och framföra synpunkter på vald metodik. Det innebär att vi kommer att argumentera för vår valda metodik "[...]korrespondens med problemställningen och dess bärkraft för de slutsatser och tolkningar som sedan görs." (Backman, 1998, s.38).

2.1 Vetenskapsteoretisk ståndpunkt

Vi kommer här att presentera det angreppssätt vi valt för vår studie. Vår vetenskapliga ansats skildrar den kunskapsteoretiska ståndpunkt vi tillämpat. Detta innebär, motiveringar till huruvida studien är kvalitativ eller kvantitativ, samt vår uppfattning om förhållandet mellan teori och empiri.

2.1.1 Vetenskaplig kunskapsuppfattning

Forskare kan ha olika uppfattningar om vad kunskap är. Vad som kan betraktas som godtagbar kunskap, beror på vilken kunskapsteori som tillämpas enligt Bryman och Bell (2003). Exempelvis påverkar valet av kunskapsteori huruvida man bör studera en social verklighet. Eftersom detta är vad vi ämnar göra, så är det aktuellt att redogöra för vår kunskapsteoretiska ståndpunkt.

För att uppfylla vårt syfte och kunna beskriva de uppfattningar som respondenterna uttrycker, kommer detta arbete att vara baserat på vår tolkning av deras verklighet. Något som anses vara en fördel för person A, kan tyckas vara en nackdel för person B. Tolkningarna kan således ses som subjektiva utifrån hur varje individ upplever situationen. Vi som undersökare kommer att tolka svaren från våra intervjuer med företag i vår urvalsgrupp. Vi kommer vidare att utifrån vår tolkning och kunskap inom området söka förståelse för hur öppen programvara uppfattas med avseende på mjukvarukvalitet. Den kunskap vi söker är av den karaktären att Wallén (1996) skulle placera in den under rubriken tolkningslära. I kvalitativa studier är tolkningsläran, det samma som hermeneutik enligt Wallén (1996). Bryman och Bell (2003) menar att hermeneutiken är en typ av forskningsstrategi som är mycket användbar där mänskligt beteende är studieobjekt. När en forskare ur en hermeneutisk ståndpunkt studerar människor och den sociala verkligheten så gör han/hon subjektiva tolkningar för att skapa en förståelse. Tolkningsläran stämmer således väl överens med vår syn på kunskap. Detta eftersom vi söker kunskap genom tolkningar av respondenters subjektiva åsikter i vår urvalsgrupp.

2.1.2 Kvantitativ och kvalitativ forskningsmetod

Det finns två huvudsakliga forskningsmetoder som vetenskapligt kan tillämpas. Vi motiverar här vårt val av kvalitativ metod. Enligt Backman (1998) skall en forskare välja en metod som korresponderar till problemformuleringens natur.

Den kvalitativa forskaren har enligt Backman (1998) ett mer subjektivt synsätt, och fokus ligger ofta på att studera hur människor uppfattar och tolkar situationer. Wallén (1996) kompletterar förståelsen av kvalitativa studier genom att förklara att de är nödvändiga för sådant som är vagt eller subjektivt, t.ex. upplevelser och känslor. Uppfattningar om öppen programvara kan vara problematiskt att mäta på grund av subjektiviteten i begreppet upplevd kvalitet. Därmed har vi, i enighet med Wallén, antagit en kvalitativ forskningsansats. Vidare så skriver Wallén att om det inte går att mäta ett fenomen som upplevelse, så får du fråga om det. Eftersom vi har studerat ett fåtal individer på ett djupare plan, snarare än att titta på många studieobjekt på ett ytligt plan, så passar en kvalitativ studie väl enligt Bryman & Bell (2003).

2.1.3 Forskningsansats

Denna uppsats kommer att ha en abduktiv forskningsansats. Detta innebär en kombination av induktion och deduktion. Den utgår från empiri, som den induktiva ansatsen, men utan att ta avstånd från teori som den deduktiva ansatsen. Teori och empiri anpassas med en växelverkan fortgående under arbetet där båda delar omformas beroende på varandra. (Alvesson & Sköldberg, 1994, enligt Fjellman & Neste, 2005). Vi utgick från empiri, dock med viss inhämtad teori innan studien påbörjades. Allt eftersom vi arbetade med empirin så inhämtades den nödvändiga teorin i relation till den insamlade datamassan.

2.2 Validitet och reliabilitet

Validitet och reliabilitet är två av de viktigaste begreppen vid evaluering av kvalitativ forskning enligt Bryman och Bell (2003).

Reliabilitet är forskningsresultatets tillförlitlighet. Detta innebär att samma resultat skall gå att uppnå igen vid en replikation av studien. Skulle man få ett annorlunda resultat vid en replikerande mätning så är forskningsresultatet mindre trovärdigt. För att uppnå en högre tillförlitlighet i resultaten är vår avsikt att inte ställa ledande eller lärande frågor om varken mjukvarukvalitet eller öppen programvara. Respondenten skall således ha sin egen bild om vad exempelvis användarvänlighet, öppen programvara och funktionalitet är för något. På så sätt påverkas respondenten i så liten utsträckning som möjligt av undersökaren. Vi har även meddelat berörda företag att de inte kommer att namnges; graden av anonymitet är således hög. Detta innebär att företagets respondenter inte känner att han/hon behöver framföra företaget i så god dager som möjligt och därför svara på ett vinklat eller felaktigt sätt.

Validitet handlar enligt Bryman och Bell (2003) om de slutsatser som genererats och om de hänger ihop, eller inte hänger ihop, med den undersökning som gjorts. Med andra ord ska de frågor man ställer till respondenten mäta det som man verkligen avser sig mäta (Mason, 1996, enligt Bryman & Bell, 2003). Det kan exempelvis tänkas att någon av våra respondenter svarar för någon annan än sig själv och då mäter vi inte längre respondentens egen uppfattning. Vederbörande kan exempelvis känna att den bör svara på ett speciellt sätt för att representera företagets kultur och värderingar. Det är därför viktigt för oss att vara medvetna

om dessa fenomen för att hålla en aktiv uppmärksamhet för liknande bias. Ett sätt att undgå oreflekterade standardiserade svar är att gå på djupet i varje fråga, enligt Bryman och Bell (2003). Respondenten får då tänka efter och ta sig tid att svara efter egna tankar och värderingar.

För att ytterligare styrka reliabiliteten och validiteten i studien fick respondenterna tillfälle att granska våra tolkningar av deras svar, för att kunna rätta eventuella feltolkningar från vår sida. Detta är något som kallas respondentvalidering av Bryman och Bell (2003). Det respondentvalideringsbrev som skickades ut återfinns i Bilaga 2.

2.3 Etik

De etiska aspekterna under denna studie har varit viktiga för oss. Framförallt för att upprätthålla institutionens goda rykte i näringslivet samt underlätta för liknande studier av efterkommande studenter. För att uppnå en god etik har följande åtgärder vidtagits, rekommenderade av Bryman och Bell (2003):

Vi informerade berörda respondenter om det riktiga syftet för undersökningen och försökte inte ge någon vilseledande information för att få företag att ställa upp på intervjuer.

Vi informerade berörda respondenter att det är OK att avsluta intervjun när helst de ville. Det gällde även deras rätt att ta tillbaka data innan rapporten publicerats.

På grund av examinationsskäl informerade vi de berörda respondenterna att rådata kan komma att granskas av examinatorer.

Vi gjorde våra respondenter medvetna om att det inte utgår någon ersättning för att delta i studien. Däremot erbjöds de ett exemplar av den färdigställda rapporten.

2.4 Genomförandet

2.4.1 Litteratursökning och kritisk granskning

Vår litteratursökning är till sin karaktär systematiskt inriktad genom att beakta de rekommendationer och faser som beskrivs av Landén (2007). För att börja den systematiska litteratursökningen behövde vi en så preciserad frågeställning som möjligt att utgå från. Medvetna om att denna kan komma att ändras under studiens gång, visste vi ändå vilka aspekter av ämnet som var centrala för en litteratursökning. Följande understrukna aspekter ansåg vi vara av intresse för vår studie.

Hur definieras mjukvarukvalitet?

Vilka uppfattningar finns det om öppen programvaras mjukvarukvalitet?

Hur ser dessa uppfattningar ut för de underattribut som tillsammans utgör helhetsbegreppet mjukvarukvalitet?

Vi letade synonymer och begrepp som täckte ovanstående begrepp för att utöka vårt urval av sökord. Sökorden användes för att söka i databaser i bland annat titel, nyckelord och abstractfälten. Sökningarna gav från första början ett väldigt stort antal träffar. Sökorden förfinades därför, för att få så precisa sökresultat som möjligt. Förfiningen av sökningarna resulterade i de sökordskombinationer som presenteras i Tabell 2-1. Här presenteras även de databaser som använts.

Tabell 2-1 – Sökord och databaser

Databas	Sökord	Antal träffar
Elin	"software quality model" AND "characteristics"	6
	ti:"software quality" AND "definition"	31
	ti:"open source software" AND "software quality"	19
	ti:"open source software" AND "usability"	7
	ti:"ISO 9126"	1
Uppsatser	"Mjukvarukvalitet"	3
	"öppen källkod"	14
Google Scholar	"open source software quality"	37

Vi gick till en början igenom sökträffarnas titlar, och utifrån detta kunde många artiklar direkt uteslutas. För att kunna avgöra vilka artiklar som kunde bidra med relevant information inom ämnet gjordes urval utifrån artiklarnas abstract. Detta första urval resulterade i en stor gallring och endast ett mindre antal artiklar, som vi ansåg vara av intresse, fanns kvar efter urvalet.

Granskningen av abstract var inte den enda metoden för att ta fram litteratur i det första urvalet. Genom att läsa igenom tidigare, för ämnet relevanta publicerade C- och D-uppsatser, rapporter från myndigheter med mera, kunde vi hitta mer litteratur inom ämnet. Denna teknik resulterade i en hel del litteratur som sedan inhandlats eller lånats från både Lund och Malmö universitet. I de fall då böckerna inte varit tillgängliga har vi i enstaka fall använt oss av sekundärkällor. Backman (1998) skriver att detta bör undvikas i en vetenskaplig rapport i den mån det är möjligt, något vi också strävat efter.

Vidare behövde vi göra ett andra urval. Detta gjordes för att exkludera de böcker och artiklar som vi inte ansåg hålla måttet för de källkritiska kriterier vi satt upp. Av samma anledning var den kritiska granskningen av litteraturen bland annat fokuserad på att ta reda på bakgrunden till hur informationen hade uppstått. Detta eftersom informationen kan vara vinklad för beställaren av forskningen. En annan del av granskningen handlade om litteraturens aktuella värde. Vi fick således alltid ställa oss frågan, "Är den här litteraturen fortfarande aktuell, finns det revideringar och nya upplagor som har relevans för vår undersökning?".

Det andra urvalet exkluderade litteratur efter bland annat ovannämnda kriterier, men också då vi insett att vissa delar av litteraturen inte var relevant för vår undersökning. Dessa slutgiltiga artiklar, böcker och rapporter som togs fram är de som användes för att som Backman (1998) skriver, sammanställa en komprimerad version av den tidigare samlade kunskapen inom området. Denna sammanställning presenterar vi i vårt teoretiska ramverk (se kap 3).

En viktig aspekt att nämna i samband med litteratursökningen är att det var en iterativ process som fortgick under hela studiens livscykel. Vi var således alltid medvetna om att vi kunde behöva gå tillbaka för att komplettera med ny litteratur.

Vi ansåg det även viktigt att vara medvetna om orsaken till en litteratursökning. Backman (1998) har identifierat en motsägelse när det gäller litteraturgranskningen och kvalitativa studier. En del forskare med en kvalitativ ansats påstår att forskaren bör gå in som teoretiskt nollställd i studien. Detta för att inte befästa tidigare teorier eller förutfattade meningar i resultatet, eftersom att forskaren "*själv helt eller delvis utgör instrumentet*" (Backman, 1998, s.51). Vi går dock in i denna studie med en teoretisk bakgrund om vad som i stora delar av forskarvärlden anses vara allmänt vedertaget när det gäller mjukvarukvalitet. Detta för att utforma en bra intervjuguide eller som Backman (1998) skriver: "*ge metodiska uppslag*" (s.51).

2.4.2 Urval av respondenter

Det val av respondenter vi gjorde är grundat på ett flertal orsaker. På grund av att vi bestämt oss för en flexibel intervjuform, ansåg vi det vara lämpligt att träffa respondenterna ansikte mot ansikte för att lättare kunna uppnå hög flexibilitet. Det blev på så vis mer personligt och det blev lättare för oss att läsa av små och subtila signaler från respondenten. Vi kunde därmed vid behov anpassa frågeställningarna efter exempelvis ansiktsuttryck och kroppsspråk, något som hade kunnat gå förlorat vid exempelvis mail- eller telefonintervjuer enligt Bryman och Bell (2003). För att kunna uppnå detta begränsade vi vårt urval geografiskt till Lund/Malmö då vi ansåg detta vara inom rimligt avstånd för oss.

Ett annat krav vi hade var att intervjua personer med antingen direkt medverkan vid, eller god insikt i, val av programvara i företag och organisationer. Detta gjordes för att vi ville vara säkra på att de personer vi intervjuade skulle ha en så god uppfattning som möjligt om de beslut som gjorts vid val av programvara.

Ytterligare ett krav var att de i sin yrkesroll någon gång övervägt ett mjukvarualternativ utvecklat med öppen källkod. Dock var det inget krav att detta övervägande skulle ha resulterat i något speciellt utfall, eftersom det var övervägandet i sig vi ville undersöka närmare. Då vi ansåg att det fanns högre sannolikhet att större företag och organisationer lagt ner mer tid och resurser på att utvärdera programvara, valde vi att inte kontakta mindre

företag då deras resurser vanligtvis är begränsade. Vi sökte således inte respondenter i en specifik bransch. Fokus för oss var snarare en specifik yrkesroll, oberoende av bransch.

Trots att materialet var litet ansåg vi det vara lämpligt att eftersträva en så stor branschmässig spridning som möjligt av våra respondenter. Vi delade in materialet i kategorierna: "privata företag" och "offentlig sektor". Inom dessa två kategorier sökte vi även där en spridning.

I kategorin "offentlig sektor" intervjuades en grundskola samt en förvaltning inom sjukvården. Vi ansåg dessa vara två vara lämpliga organisationer eftersom de utgör stora områden inom den offentliga sektorn. Vidare är det av intresse att de valda organisationerna, lyder under lagen om offentlig upphandling (LOU). Enligt Expowera (2007) innebär LOU, att all offentlig upphandling måste ske i konkurrens, vara affärsmässig och ske på ett objektivt sätt. På så vis "garanteras" att dessa organisationer haft ett vitt perspektiv när de upphandlat programvara.

De två privata företag vi intervjuat verkar inom IT- och finansbranschen. Detta val gjordes därför att vi hade uppfattningen att dessa två typer av företag troligtvis skulle ha skilda åsikter om vad de ansåg viktigt i en programvara. Banker eftersträvar hög säkerhet i programvaran och tar det säkra före det osäkra. IT-företag ligger däremot ofta i framkant med tekniska lösningar och söker därmed alternativa och nya vägar.

De företag och organisationer vi valde, gjordes utifrån de kriterierna som är listade ovan. Utifrån detta gjorde vi vad Bryman och Bell (2003) kallar ett bekvämlighetsurval, där vi ringde runt till företag med lokal förankring vi sedan tidigare hade kännedom om.

2.4.3 Intervjuer

En intervju kan enligt Bryman och Bell (2003) vara antingen strukturerad, semistrukturerad eller ostrukturerad. Då vi sökt att få en bild av uppfattningar behövde vi utföra våra intervjuer under relativt öppna förhållanden för att respondenten skulle kunna uttrycka och förklara sig fritt. För att kunna göra detta skall respondenten enligt Bryman och Bell (2003), ges utrymme att gå utanför en eventuell intervjuguide. Innan intervjuerna gjorts kunde vi omöjligt veta vad varje respondent tycker är viktigt. Därmed kunde vi inte skapa en passande intervjuguide för varje person. Bryman och Bell (2003) skriver att kvalitativa intervjuer tenderar att vara mer följsamma i den riktning som respondenten svarar. Det stod således klart att vi behövde en flexibel intervjumetod för att kunna genomföra en studie där vi tar reda på de uppfattningar som finns kring öppen programvara avseende mjukvarukvalitet.

Hur flexibel skulle vår intervjuguide vara? Bryman och Bell (2003) skriver att: "*Om forskaren inleder sin undersökning med förhållandevist tydligt fokus [...] kommer han eller hon sannolikt att använda sig av semistrukturerade intervjuer*" (s.366). Samtidigt önskade vi att ha en viss struktur på intervjuguiden, för att säkerställa en jämförbarhet mellan olika intervjuer och respondenter. Vi ansåg oss ha ett "förhållandevist tydligt fokus" med vår undersökning, då vi sökte uppfattningar om ett antal väl definierade kvalitetsattribut. En relativt öppen intervjuform gav oss också möjlighet att diskutera kvalitetsattribut med respondenten. På så vis kunde vi säkerställa att de svar vi erhöll verkligen syftade till de frågor vi ställt. Det stod således klart att Bryman och Bells (2003) rekommendationer för en semi-strukturerad intervjuform överensstämde väl med de kriterier vi hade för datainsamlingen.

Eftersom vi utformade frågorna efter de kvalitetsattribut som finns beskrivna i ISO/IEC 9126:1991 så speglas uppsatsens fokus väl. Intervjufrågorna återfinns i bilaga 3.

För att utföra intervjuer med god kvalitet studerade och använde vi oss av Kvale's (1996, enligt Bryman & Bell 2003) tio kriterier för en framgångsrik intervjuare. Dessa tio punkter visade sig vara mycket användbara för oss, framförallt med tanke på att vi sedan tidigare endast hade begränsad erfarenhet av intervjuförfarande. Samtliga intervjuer tog plats på respondenternas arbetsplats för att de skulle känna sig trygga i miljön. Att utföra intervjuer ute på arbetsplatsen bidrog även till att vi fick bekanta oss med den miljö där intervjupersonen arbetar, vilket enligt Bryman och Bell (2003) underlättar för tolkning av, och förståelse för, de svar som ges under intervjun.

De personer som accepterade att ställa upp på en intervju fick ta del av ett introduktionsbrev från oss via e-mail. I detta beskrevs kort syftet med intervjun och de ämnen intervjun skulle komma att handla om. Brevet skickades ut för att respondenterna skulle få möjlighet att reflektera i god tid över ämnet. På så vis blev de svar vi erhöll mer genomtänkta. Detta brev finns i en anonymiserad version i Bilaga 1.

2.4.4 Transkriberingar

Bryman och Bell (2003) skriver att det är viktigt att de som intervjuar och transkriberar intervjuerna har samma slags erfarenhet och utbildning. Detta för att undvika att information skall gå förlorad eller misstolkas i transkriberingsarbetet. För att minska denna risk var båda författarna av denna uppsats, delaktiga både under intervjuerna samt i transkriberingen.

Vid slutet av varje intervju påbörjades en transkriberingsprocess. Transkriberingen från varje intervju finns i Bilaga 4. I transkriptionsprotokollen återfinns följande meningar inom klamrar på ett fåtal ställen:

[Här väljer vi att inte transkribera en mening då det anses vara känslig information, och ej heller berör uppsatsens ämne]

På dessa ställen har respondenten yttrat sig om något som vi valt att inte transkribera. Detta för att garantera att känslig information ej sprids vidare.

[Företagsnamn]

Företagsnamnet utelämnas för att säkerställa anonymitet.

[Ohörbart]

På grund av ljudtekniska skäl förekommer denna notering ett fåtal gånger. Detta beror på störningar i närheten under intervjun.

2.4.5 Analys av intervjuer

Analys av insamlad data är, enligt Backman (1998), den svåraste delen av den kvalitativa forskningsprocessen. För att underlätta detta arbete tillämpade vi ett systematiskt och metodiskt tillvägagångssätt. Det är genom denna process vi organiserat, tolkat och strukturerat den mängd data vi erhållit via intervjuerna.

I enlighet med Bryman och Bell (2003), så började vi vår analys av data med kodning. Detta gjordes redan innan vi var färdiga med samtliga intervjuer. På så vis minimerade detta känslan av att insamlade data kunde upplevas som en alltför stor massa. Första steget i kodningen var genomläsning av transkriberingarna från intervjuerna, vilket gav oss en överblick över det material vi skulle hantera. Därefter gjordes en noggrannare genomgång av var och en av transkriberingarna. I texten markerades, med hjälp av kodord, i enlighet med Bryman och Bell (2003) (se Tabell 2-2), delar av de svar som var av intresse för uppsatsens syfte. Kodorden var på förhand valda utifrån från den litteratur som låg till grund för vår intervjuguide (ISO/IEC 9126:1991), exempelvis ”mognad”, ”säkerhet” och ”interoperabilitet”. För att tydliggöra för läsaren presenteras nedan ett exempel på hur en sådan kodning såg ut:

Meningen: "*Och, ehh, för våran del, så kan vi bara förlita oss på det som är väl beprövat. Vi tänker inte gå först i något håll i vad det gäller något område, inte säkerhet heller.*", kodas och grupperas med "Säkerhet" för att detta lätt skall kunna paras ihop med andra uppfattningar respondenten har om säkerheten i öppen programvara.

Eftersom intervjun är semistrukturerad till sin karaktär så tilläts respondenten att tala utanför de direkta frågorna. Detta medförde att en del av svaret på en fråga om exempelvis programvarans "mognad", fick kodas som "säkerhet", då respondenten stundtals svävat iväg och pratat om annat än vad som efterfrågats. Även detta talar för vikten av att koda materialet för att styrka validiteten i studien.

Vi har försökt begränsa subjektiviteten i våra tolkningar, i den mån det varit möjligt. Detta har gjorts genom att vi två, oberoende av varandra, har tolkat innebörden i de svar vi erhållit från respondenterna. Därefter har vi gått igenom tolkningarna gemensamt. På de punkter där tolkningarna varit olika, har vi diskuterat och resonerat fram en slutlig tolkning. Detta görs för att uppnå hög nivå, av vad Bryman och Bell (2003) kallar ”interkodarreliabilitet”.

När denna första del av kodningen var avklarad, kategoriserade vi de kodade meningarna under sex olika teman. Detta gjordes för att få datamassan mer lättöverskådlig. Dessa teman hade även de sin grund i ISO/IEC 9126:1991, exempelvis ”funktionalitet”, ”flyttbarhet” och ”underhållsmässighet”. Datan användes som en del av grunden till analysdelen, som presenteras senare i rapporten.

Tabell 2-2 - Koder och teman

Tema	Kod
Funktionalitet	<i>Ändamålsenlighet</i>
	<i>Noggrannhet</i>
	<i>Interoperabilitet</i>
	<i>Säkerhet</i>
Tillförlitlighet	<i>Mognad</i>
	<i>Feltolerans</i>
	<i>Återhämtning</i>
Användbarhet	<i>Begriplighet</i>
	<i>Lärbarhet</i>
	<i>Körbarhet</i>
	<i>Attraktivitet</i>
Produktivitet	<i>Tidsbeteende</i>
	<i>Resursutnyttjande</i>
Underhållsmässighet	<i>Analyserbarhet</i>
	<i>Föränderlighet</i>
	<i>Testbarhet</i>
	<i>Stabilitet</i>
Flyttbarhet	<i>Anpassbarhet</i>
	<i>Installerbarhet</i>
	<i>Ersättningsbarhet</i>

2.5 Språkliga reservationer

Uppsatsen framläggs på svenska och vår ambition har varit att i möjligaste mån försvenska de facktermer som används. Mycket av den litteratur vi studerat har varit skriven på engelska. För att hålla oss till så vedertagna översättningar som möjligt, har vi sökt upp vetenskapliga artiklar och uppsatser på svenska som behandlat samma begrepp och använt oss av deras terminologi.

3. Teoretisk referensram

Detta kapitel syftar till att:

(1) Beskriva, definiera och tydliggöra de begrepp och teorier som används i studien för att klargöra den problematik som står i relation till begreppet mjukvarukvalitet.

(2) Presentera en modell som representerar de attribut forskarvärlden satt upp för mjukvarukvalitet. Modellen som presenteras kommer att ligga till grund för att ställa relevanta frågor kring mjukvarukvalitet i våra intervjuer. Avsnittet kommer att inledas med en kortare historik kring begreppen kvalitet och mjukvarukvalitet för att styrka trovärdigheten för valet av vår modell.

(3) Presentera en sammanställning av vetenskapliga granskningar avseende mjukvarukvalitet i öppen programvara. Den bild av vad som tidigare publicerats i ämnet kommer vi att använda oss av i analysen, för att identifiera likheter och skillnader i de observationer vi gjort i form av intervjuer.

3.1 Produkt- och Mjukvarukvalitet

Mjukvarukvalitet står i fokus för denna undersökning och vi behöver därmed också reda ut begreppet. Côté et al (2007) skriver att "Software Quality Engineering" är en disciplin inom mjukvaruutveckling som syftar till att förbättra kvalitén i mjukvara. Vidare hävdar samma författare, att denna disciplin behöver utgå och enas om en kvalitetsmodell som täcker alla olika användares definitioner av kvalité. En mjukvarukvalitetsmodell försöker tydliggöra kvalitetsbegreppet, enligt Håkansson (2000). Det står således klart att vi, precis som Côté et al (2007), söker en sådan mjukvarukvalitetsmodell som tar hänsyn till samtliga respondenters syn på av kvalitet.

Definition av produkt- eller mjukvarukvalitet har alltid varit en debatterad fråga enligt Côté et al (2007). Anledningen till detta är, enligt Håkansson (2000), att det oftast finns olika sorters intressenter av en produkt med olika syn på vad kvalité är för dem. De har därmed också olika syn på hur kvalité bör definieras. I domäner som mjukvaruutveckling kan de olika intressenterna exempelvis vara slutanvändare, marknadsförare eller programmerare. Intressenterna bör, enligt Côté et al (2007), identifieras och kategoriseras efter olika kvalitetsperspektiv för att man ska kunna nå konsensus kring en kvalitetsmodell.

De olika perspektiven som Côté et al (2007) syftar på är sammanfattade av Garvin (1984). Garvin syftade från början på produktkvalitet och inte på mjukvarukvalitet, men Kitchenham och Pfleeger (1996) applicerar samma kvalitetsperspektiv på mjukvaruindustrin, och kallar det därmed för mjukvarukvalitet.

Här följer olika mjukvarukvalitetsperspektiv sammanfattade av Kitchenham och Pfleeger (1996) från Garvin (1984):

- Transcendentala perspektivet
- Användarperspektivet
- Tillverkningsperspektivet
- Produktperspektivet
- Det värdebaserade perspektivet

För att nå konsensus kring en definition av mjukvarukvalitet så bör forskare hitta en modell som tar hänsyn till alla olika perspektiv för mjukvarukvalitet enligt Cote et al (2007). Med vetskap om att det är de olika intressenterna som påverkar vilka kvalitetsperspektiv som bör tas hänsyn till i det specifika fallet, så bör vi enligt Håkansson (2000) identifiera våra intressenter i studien. Det är således för oss ointressant om en modell täcker de fem perspektiv som beskrivs av Kitchenham och Pfleeger (1996). Det är dock av yttersta intresse för oss att modellen täcker de perspektiv som representerar våra respondenter. Samtidigt skall modellen vara allmänt accepterad av forskarsamhället.

3.1.1 Respondenternas kvalitetsperspektiv

Vi syftar i detta avsnitt att klargöra vilka kvalitetsperspektiv som är relevanta för vår studie. I vår undersökning kommer som bekant primärdata att samlas in i form av intervjuer från personer med sådana yrkesroller där de har inflytande över val av programvara i en organisation eller ett företag. Enligt rapportens avgränsningar så ämnar vi endast studera de mjukvaror som respondenterna själv använder och inte de som de eventuellt utvecklar i sitt yrke. Således kommer tillverkningsperspektivet inte att vara aktuellt i vår studie. Inte heller kommer det transcendentala perspektivet behöva täckas av vår modell. Vi söker som bekant en modell för att definiera mjukvarukvalitet. Det transcendentala perspektivet visar däremot på att kvalitet inte är definierbart, dock igenkännbart. Côté et al (2007) återger en beskrivning där aspekterna på det transcendentala perspektivet faktiskt går att uppnå med en holistisk ansats. För dem som är intresserade kan mer information hittas i Côté et al (2007).

Här följer en kort beskrivning, samt ett par exemplifierande citat för att visa vad som ligger till grund för kvalitetskategoriseringar. Följande information är hämtad ur Kitchenham och Pfleeger (1996) och Garvin (1984).

Användarperspektivet, precis som de andra perspektiven, utvärderar en viss aspekt av qualité. Det är väldigt konkret och personligt ur den synvinkeln att det utvärderar produktens egenskaper för att uppnå det definierade målet för den specifika användaren. Även mjukvarans användbarhet i relation till produktens funktioner ses som centralt i denna kategorisering. (Kitchenham & Pfleeger, 1996; Garvin, 1984)

- *"Quality consists of the capacity to satisfy wants..."* Edwards (1968, enligt Garvin 1984)

- *"Quality is the degree to which a specific product satisfies the wants of a specific consumer"* Gilmore (1974, enligt Garvin 1984)

Produktperspektivet gör avstamp i produktens inre kvalitet, den aspekt av kvalitet som exempelvis upplevs av programmeraren. Litteraturen har beskrivit förhållandet där den inre kvaliteten påverkar den yttre kvalitén, exempelvis den som upplevs i användarperspektivet. Det intressanta här är att den inre kvaliteten kan bedömas objektivt och är kontextuellt oberoende. Kvalitén i produkten är direkt representerad av kvantiteten i ett attribut hos produkten. Exempelvis så är antalet (kvantiteten) felfria kodrader (attribut) i en programvara direkt representativt för produktens kvalitet. (Kitchenham & Pfleeger, 1996; Garvin, 1984)

- *"Differences in quality amount to differences in the quantity of some desired ingredient or attribute"* Abbott (1953, enligt Garvin 1984)

Det värdebaserade perspektivet används för att kunna utvärdera produktens kvalitet i relation till kostnader. En bra kvalitetsmodell behöver exempelvis kunna hantera utvärdering av krav och tillåta revideringar av dessa för att främja en mindre kostnad av mjukvaran. Det skall således vara möjligt att göra avvägningar mellan kvalitet och kostnad. (Kitchenham & Pfleeger, 1996; Garvin, 1984)

- *"Quality means best for certain customer conditions. These conditions are (a) The actual use and (b) the selling price of the product"* Feigenbaum (1983, enligt Garvin 1984).

3.1.2 Mjukvarukvalitetsmodeller och dess överensstämmelse med kvalitetsperspektiv

Med en medvetenhet om vilka perspektiv som nu bör täckas av vår modell söker vi bland befintliga modeller som överensstämmer med våra krav. Kvalitetsmodellerna är inget nytt fenomen, utan har diskuterats under en lång tid tillbaka. Ett exempel på en tidig kvalitetsmodell är den som McCall (1977) presenterade. Senare har även andra kvalitetsmodeller presenterats av bland annat Boehms et al. (1976 & 1978, enligt Côte et al 2007) och Dromey (1995, enligt Côte et al 2007).

Dessa modeller har testats av Côte et al (2007) efter bland annat de kvalitetsperspektiv som tidigare nämnts. I denna studie kommer författarna fram till att McCall's, Boehms et al's och Dromey's ramverk alla favoriserar produktperspektivet på bekostnad av Garvin's (1984) övriga perspektiv som vi ovan specificerat som viktiga.

Håkansson (2000) framhäver, precis som Côte et al (2007) behovet av en enad syn på kvalitetsbegreppet, och därmed en modell som kan förklara det. Detta har enligt Håkansson (2000) *"resulterat i en standard"* (s.16). Denna standard presenterades 1991 av "The International Organisation for Standardisation" och namngavs "ISO/IEC 9126 (1991): Software product evaluation - Quality characteristics and guidelines for their use". Vidare så är detta den enda modell som enligt Côte et al (2007) lämpar sig eftersom den stödjer alla de perspektiv som presenterats av Kitchenham och Pfleeger (1996) samt Garvin (1984). Även Bazzana, Andersen och Jokela (1993) skriver, att ISO/IEC 9126-standarderna ger begreppet mjukvarukvalitet en definition, som bör representera slutet av en lång diskussion som förts av forskare inom ämnet.

Med ovanstående argumentation anser vi därmed att ISO/IEC 9126 utgör en utmärkt modell för att definiera mjukvarukvalitet, av samma anledning kommer denna också att användas i studien. Den valda mjukvarukvalitetmodellens olika attribut kommer att presenteras i nästa avsnitt.

Vi anser dock att det är viktigt att bringa ljus över de synpunkter och kritik som framförts angående den standard vi valt som modell. Detta för att ha ett opartiskt förhållningssätt till de modeller som finns att tillgå. Pflieger (2001, enligt Côte et al 2007) identifierar ett par viktiga problem med ISO/IEC 9126:1991. ISO kommittén utformade därefter en reviderad modell som tar hänsyn till den framförda kritiken.

Den nya modellen ISO/IEC 9126-1:2001a är resultatet av den kritik som förts fram av forskarsamhället. Revisionen har neutraliserat de oenigheter som funnits enligt Côte et al (2007). Vår egen granskning av Côte et als kortfattade presentation av den reviderade modellen visar på ett nytillkommet underattribut. Olyckligtvis har den reviderade modellen ej kunnat användas på grund av att litteratur saknas. Därför finns ej det nya underattributet med i denna studie, något vi också diskuterar i slutet av denna rapport.

3.2 Kvalitetsattribut

Nedan följer en redogörelse för de attribut som är del av den standard för mjukvarukvalitet som beskrivs i ISO/IEC 9126:1991. I början av varje underrubrik presenteras en översättning gjord av Håkansson (2000) av de sex huvudattribut som utgör mjukvarukvalitet. Därefter följer en uppräknig, samt kortare beskrivning, av de underattribut som huvudattributen delas upp i. Uppdelningen i underattribut konkretiserar respektive attribut, vilket underlättar vårt bruk av dem senare i uppsatsen.

3.2.1 Funktionalitet

"Funktion (Funktionalitet): en mängd av attribut som har sin grund i en uppsättning funktioner och deras specificerade egenskaper. Funktionerna är de som tillgodoser uttalade eller underförstådda behov. [ISO/IEC 9126:1991]" (Håkansson, 2000, s.17).

Enligt Blomquist (2000) så innebär funktionalitet att mjukvaran ska innehålla alla de funktioner som krävs för att mjukvaran på ett tillfredställande sätt ska kunna utföra de uppgifter den är avsedd att kunna hantera. Beskrivningen av funktionalitet behöver förtydligas för att lättare kunna förstås. Nedan följer de fyra underattribut, beskriva i ISO/IEC 9126, som sammantaget utgör huvudattributet funktionalitet enligt Blomquist (2000).

- Ändamålsenlighet: funktionernas lämplighet avseende uppgiften.
- Noggrannhet: funktionernas förmåga att återge ett efterfrågat resultat.
- Interoperabilitet: programvarans förmåga att samverka med andra system.
- Säkerhet: systemets förmåga att säkerställa att ingen obehörig får tillgång till program och data.

3.2.2 Tillförlitlighet

"Tillförlitlighet: en mängd av attribut som gör att programvaran kan upprätthålla sin prestandanivå under givna villkor för en given tidsperiod. [ISO/IEC 9126:1991]" (Håkansson, 2000, s.17).

Pressman (2001) hävdar att tillförlitlighet utan tvekan är en viktig del av en programvaras totala kvalitet. Han menar att ifall ett program vid upprepade tillfällen brister i exekvering så spelar det mindre roll om resterande kvalitetsfaktorer är acceptabla. Tillförlitlighet i ISO/IEC 9126, enligt Blomquist (2000), delas in i underattributen:

- Mognad: den frekvens av avbrott som uppstår på grund av fel i programvaran.
- Feltolerans: systemets förmåga att upprätthålla en tillfredställande nivå av funktionalitet trots uppstådda fel.
- Återhämtning: den tid och kraft som krävs för att systemet efter en krasch skall kunna återställa påverkad data till stadiet före kraschen.

3.2.3 Användbarhet

"Användbarhet: en mängd av attribut som avgör den prestation som är nödvändig för användning, och på den enskilda utvärderingen av en sådan användning, av givna eller en underförstådd grupp användare. [ISO/IEC 9126:1991]" (Håkansson, 2000, s.17-18)

Nielsen (1993) hävdar att 48 % av koden i modern programvara står för användargränssnitt. Trots att siffrorna är aningen gamla så ger de ändå en indikation på att systemutvecklare genom åren lagt ner stor möda på att eftersträva god användbarhet. Pressman (2000, enligt Abrahamsson & Furu 2004) skriver att användbarhet är av en sådan stor betydelse, att ifall den är undermålig i ett program, så spelar det ingen roll ifall en fullgod funktionalitet finns att tillgå; programmet är ändå dömt att misslyckas. Enligt Blomquist (2000) delas användbarhet upp i

- Begriplighet: den prestation som krävs av användaren för att denne ska kunna förstå den logiska strukturen i programvaran.
- Lärbarhet: ansträngningen som krävs av användaren för att denne ska lära sig hur programvaran tillämpas.
- Körbarhet: avser användarens ansträngning för hantering och operationskontroll i programmet.
- Attraktivitet: programvarans förmåga att tilltala användaren.

3.2.4 Produktivitet

"Produktivitet (effektivitet): en mängd av attribut som beror på förhållandet mellan det som programvaran utför och mängden av resurser som under givna villkor använts. [ISO/IEC 9126:1991]" (Håkansson, 2000, s.18)

Blomquist (2000) skriver att effektivitet hos en programvara framför allt märks i den tid det tar för programmet att utföra en viss handling. Inte minst märks detta i realtidssystem, men även i vanliga applikationer kan dålig effektivitet leda till förseningar vilket kan medföra att kvalitén upplevs som undermålig. Detta attribut delas upp i följande underattribut enligt Blomquist (2000):

- Tidsbeteende: svars- och bearbetningstider för att funktioner skall exekveras.
- Resursutnyttjande: mängd resurser i form av annan mjukvara, hårdvara, material samt personal som krävs för att en funktion skall utföras.

3.2.5 Underhållsmässighet

"Underhållsmässighet: en mängd av attribut som avgör vilka resurser som är nödvändiga för att utföra specificerade ändringar. [ISO/IEC 9126:1991]" (Håkansson, 2000, s.18)

Pressman (2001) skriver att underhåll av programvara är den aktivitet av systemutveckling som tar mest tid i anspråk. Underhållsmässighet är ett mått på med vilken lätthet ett system kan korrigeras ifall: ett fel uppmärksammas, omgivningen förändras eller användarna uttrycker förändringar i krav på systemet. Underattributen för underhållsmässighet i ISO/IEC 9126 är enligt Blomquist (2000):

- Analyserbarhet: arbetsinsatsen som krävs för att analysera brister och felkällor samt identifiering av delar i programmet som behövs bytas ut.
- Föränderlighet: det arbetet som krävs för att modifiera programvaran.
- Testbarhet: arbetsinsats som krävs för att validera modifierad programvara.
- Stabilitet: mått på i vilken utsträckning modifieringar av programvara riskerar att påverka oönskade delar av programmet.

3.2.6 Flyttbarhet

"Flyttbarhet: en mängd av attribut som avgör vilka resurser som krävs för att flytta programvaran från en miljö till en annan. [ISO/IEC 9126:1991]" (Håkansson, 2000, s.18)

Blomquist (2000) skriver att dessa attribut ger en indikation på hur flyttbar programvaran är. Detta gäller inte bara plattformsoberoende utan flyttbarhet gäller även miljöer i en större betydelse såsom olika mjuk/hård-vara samt mellan/inom organisationer. Underattributen för flyttbarhet är som följer enligt Blomquist (2000):

- Anpassbarhet: att med endast tillhandahållna hjälpmedel avsedd för programvaran kunna anpassa den till en specifik miljö.
- Installerbarhet: den ansträngning som krävs för att installera programvaran i en viss miljö.
- Ersättningsbarhet: i vilken utsträckning det är möjligt, samt vilken ansträngning som krävs, för att programvaran skall kunna ersätta annan programvara som är befintlig i en specifik miljö.

3.3 Forskningens syn på mjukvarukvalitet i öppen programvara

Litteraturen som behandlas här, uttrycker vetenskapligt grundade åsikter om mjukvarukvalitet i öppen programvara. Dessa åsikter kategoriseras efter de huvudattribut som presenterats i föregående avsnitt i enlighet med ISO/IEC 9126:1991. Detta görs för att vi senare skall kunna jämföra forskningens syn med de uppfattningar vi inhämtat via det empiriska arbetet. Teorin i detta kapitel är till huvudsak baserad på Samoladas och Stamelos (2003) som samlat ihop en mängd vetenskapliga studier om öppen programvara och dess mjukvarukvalitet och sammanställt dessa. Vår egen litteratursökning syftar till att identifiera studier som är gjorda efter 2003, då Samoladas och Stamelos (2003) publicerade sin artikel. Vår egen litteratursökning bör således endast betraktas som ett komplement till Samoladas och Stamelos (2003).

Låt oss först påpeka att det inom detta område, som så många andra, råder en debatt om vem som har "rätt" och vem som har "fel". Framförallt handlar det om förespråkare, respektive motståndare, till öppen programvara. Det vill säga, det finns en mängd olika åsikter vi identifierat. Vår sammanställning har för avsikt att vara så opartisk som möjligt då vi granskat ett spektrum av åsikter. Vi återger de som vi anser vara representativa för det material vi gått igenom.

Både Raghunathan, Prasad, Mishra och Chang (2005) och Samoladas och Stamelos (2003) skriver att debatten om kvalitén i öppen programvara lider av en tydlig avsaknad av bevis. Genom att sammanställa tidigare empiriska studier och vetenskapliga artiklar anser Samoladas och Stamelos (2003) att de framställer vetenskapliga bevis, för hur mjukvarukvalitén i öppen programvara, de facto är.

3.3.1 Forskningens syn på funktionalitet

- Ändamålsenlighet
- Noggrannhet
- Interoperabilitet
- Säkerhet

Samoladas och Stamelos (2003) skriver, utifrån sin studie, att det samlade begreppet funktionalitet i öppen programvara är förhållandevis hög.

Det finns enligt Samoladas och Stamelos (2003) domäner där öppen programvara fortfarande är det enda alternativet för efterfrågad funktionalitet. Det är således endast öppen programvara som har funktionalitet som är så pass "ändamålsenlig" att den går att använda i dessa sammanhang. Dessa exempel täcker framförallt områden som operativsystem och nätverksapplikationer. Samoladas och Stamelos (2003) förespråkar vidare forskning för "ändamålsenligheten" samt "noggrannhetens" kvalitet inom mer vardagliga applikationer. Det är därför svårt att dra en slutsats om öppen programvaras generella kvalitet avseende dessa attribut. Även Raghunathan et al. (2005) instämmer i svårigheten att bedöma funktionalitetens kvalitet på ett generellt plan. Raghunathan et al. (2005) argumenterar dock för att den generella öppna programvarans funktionskvalité kan ses som likvärdig med den slutna programvaran.

Ett ständigt införande av nya standarder är en av orsakerna som bidrar till den höga "interoperabiliteten" i öppen programvara enligt Samoladas och Stamelos (2003). I proprietär programvara kan detta begränsas av att leverantören tar betalt för införandet av stöd för exempelvis nya filsystem. I öppen programvara får vem som helst vara med och bidra med idéer och lösningar.

Hoepman och Jacobs (2007) behandlar säkerhetsaspekten i programvara; en annan aspekt av funktionaliteten. I deras artikel ställs öppen programvara mot slutna programvara. Argumentationen mynnar ut i ställningstagandet att öppen programvara uppnår högre "säkerhet". Detta motiveras genom att ju fler personer som aktivt deltar i att leta upp säkerhetsluckor, desto större är chansen att identifiera och rätta till dessa brister. Öppen programvara låter även användaren själv utvärdera säkerheten i ett program, eller om man så önskar, låta en tredje part utföra arbetet. På grund av avsaknaden av tillgång till källkoden är detta inte möjligt i samma utsträckning med slutna programvara. Dock hävdar Hoepman och Jacobs (2007) att öppen programvara initialt, då källkoden publiceras, är väldigt exponerad för hot då illasinnade personer får tillgång till den. I det långa loppet hävdar Hoepman och Jacobs (2007) att man uppnår en sådan hög "säkerhet", att den initiala risken är väl värd att ta.

3.3.2 Forskningens syn på tillförlitlighet

- Mognad
- Feltolerans
- Återhämtning

Paulson, Giancarlo och Armin (2004) hävdar att utveckling med öppen källkod bidrar till att buggar i programvara upptäcks tidigare. På så vis kan de rättas till snabbare vilket i sin tur innebär en mer ”mogen” applikation med mindre fel. Det beror på att det i denna utvecklingsform finns fler testare, då många av användarna också fungerar som testare. Användarna har även möjligheten i öppen programvara att identifiera defekter. De kan till och med komma med förslag på lösningar till problemet. Precis som Paulson et al. (2004) så instämmer Samoladas och Stamelos (2003) som citerar Raymond (2002) ”*Given enough eyeballs, all bugs are shallow*” (s.10).

Samoladas och Stamelos (2003) redogör inte för ”feltolerans” och ”återhämtning” i öppen programvara. Inte heller har vår kompletterande litteratursökning kunnat identifiera vetenskapligt grundade teorier på sådan detaljnivå.

3.3.3 Forskningens syn på användbarhet

- Begriplighet
- Lärbarhet
- Körbarhet
- Attraktivitet

En av de empiriska undersökningar som inkluderas i Samoladas och Stamelos (2003) vetenskapliga sammanställning har studerat användarvänligheten i öppen programvara. Det framgår att svårigheten att utföra en uppgift i den öppna programvarumiljön kan jämföras med svårigheten i slutna programvara. Svårigheten att utföra en uppgift kan jämföras med vad som i ISO/IEC 9126 kallas ”körbarhet”.

I samma studie dras även slutsatserna att det öppna programvaru-alternativet saknar en klar och enkel struktur. Underattributet för ”begriplighet” ses därmed som mindre kvalitetsmässigt i öppen programvara.

Även ”lärbarheten” i öppen programvara behandlas i samma studie. Samoladas och Stamelos (2003) drar slutsatser som visar att ”lärbarheten” i öppen programvara är hög. De drar också slutsatserna att ett byte från slutna till öppen programvara, med tanke på ”lärbarheten”, skulle vara relativt enkelt.

Det redogörs aldrig för attributet attraktivitet i några av de vetenskapliga artiklar som har granskats.

3.3.4 Forskningens syn på produktivitet

Samoladas och Stamelos (2003) hävdar att det råder total avsaknad av vetenskapligt framtagna studier av produktivitet i öppen programvara. I vår litteratursökning har vi inte heller kunnat hitta någon sådan vetenskaplig studie. Därför anser vi att forskningen inte har någon åsikt om detta kvalitetsattribut.

3.3.5 Forskningens syn på underhållsmässighet

- Analyserbarhet
- Föränderlighet
- Testbarhet
- Stabilitet

De studier som finns, visar inte på tillräckligt klara resultat i favör för vare sig öppen eller sluten programvara. Därför går det inte att dra några slutsatser om vilken typ som har högst underhållsmässighet. Även om forskningen har svårt att urskilja och enas om vilken typ av programvara som håller högst kvalitet går det trots allt att dra slutsatser om öppen programvara kan uppvisa en acceptabel nivå eller ej; allt enligt Samoladas och Stamelos (2003).

För att undersöka detta faktum sammanställde dessa forskare ett par studier angående underhållsmässigheten. Sammanställningen påvisar en tudelad bild, men en majoritet säger att underhållsmässigheten i öppen programvara är god. Samoladas och Stamelos (2003) hävdar i enlighet med Raghunathan et al (2005), att helhetsbegreppet ”underhållsmässighet” i öppen programvara håller en acceptabel nivå. Den skiljer sig inte alltför mycket från liknande studier som är gjorda avseende den proprietära programvaran.

Vid en kritisk granskning av Samoladas och Stamelos (2003) sammanställning av de empiriska studierna finner vi endast stöd för två av de fyra underattribut som utgör helhetsbegreppet ”underhållsmässighet”. ”Föränderligheten” visar sig hålla hög kvalitet, medan vad som i ISO/IEC 9126 avses med ”stabilitet” anses vara en svag länk i öppen programvara. En del av den dåliga ”stabiliteten”, har enligt samma författare, grund i en hög koppling i källkoden. Ändrar du en sak, är risken stor att en annan programfunktion berörs av modifieringen.

Laplante et al (2007) argumenterar för att underhåll av öppen programvara är lättare än i sluten, inte främst för att källkoden finns tillgänglig, utan även för att de som är med och utvecklar öppen programvara lägger ned extra ansträngning för en hög kvalitet i just detta attribut. Detta för att andra utvecklare lätt skall kunna sätta sig in i programkoden. Vidare hävdar Laplante et al (2007) vikten av en viss skepsis mot testning i öppen programvara. Utvecklingen av öppen programvara tenderar att omfatta lite testning av utvecklarna. Istället förlitar de sig på att användarna själva utföra stora delar av testningsarbetet.

Sammantaget finner vi därmed endast vetenskapligt grundade bevis för tre av de fyra underattributen i kategorin. Som tidigare nämnts anses ”föränderligheten” som kvalitetsmässigt hög. Den ansträngning som krävs för att testa och validera en öppen programvara ses i vetenskapliga sammanhang som problematiska. Även ”stabiliteten” anses hålla låg kvalitet i öppen programvara. Det sistnämnda, ”analyserbarheten”, har således varken vi eller Samoladas och Stamelos (2003), kunnat identifiera i en vetenskaplig kontext.

3.3.6 Forskningens syn på flyttbarhet

- Anpassbarhet
- Installerbarhet
- Ersättningsbarhet

Samoladas och Stamelos (2003) skriver att, ända sen begynnelsen, har ”flyttbarheten” varit en central fråga för den öppna programvaran. Vidare argumenterar dessa författare för att öppen programvaras förmåga att leverera hög flyttbarhet grundar sig i tillgången till källkoden. Som bekant handlar flyttbarheten i stort om att kunna flytta mjukvara, från en existerande hårdvarukonfiguration, till en annan. Med tillgång till källkoden, hävdar Samoladas och Stamelos (2003), att man alltid kan kompilera¹ denna, på varje unik hårdvarukonfiguration och på så sätt, få den att fungera överallt. Exempelvis anses Linux²-kärnan vara enkelt flyttbar för att källkoden är tillgänglig.

En annan aspekt av flyttbarhet är den ansträngning som krävs för installation av mjukvara. Samoladas och Stamelos (2003) hävdar att installationsförfarandet för öppen programvara länge dragits med en del problem. Framförallt betonas avsaknaden av en standard som följs vid installationer av öppen programvara. Exempelvis har installationen av Linux länge krävt att användaren skall behärska en rad parametrar, samt vara tekniskt kunnig, för att installationen skall gå att utföra. För att underlätta installationen finns dock dokumentation på Internet tillgänglig. Enligt Samoladas och Stamelos (2003) är utvecklarna medvetna om denna problematik, och har de senaste åren arbetat aktivt med att förenkla installation av öppen programvara.

Forskningsresultat om ”ersättningsbarhet” i öppen programvara redogörs ej för av Samoladas och Stamelos (2003). Inte heller har vi i vår kompletterande litteraturgranskning kunnat finna någon sådan studie.

¹ **Kompilera** – översätta (program) till direkta datorinstruktioner (Nationalencycledin, 2008)

² **Linux** – operativsystem för persondatorer (Nationalencycledin, 2008)

4. Undersökningen

Kapitlet består av två delar. Den första delen är till för att ge en bakgrund kring respondenterna och de företag/organisationer de arbetar på. Detta görs för att läsaren skall få en förståelse för de sammanhang data inhämtats från. Den senare delen innehåller en sammanställning av de svar vi erhållit via intervjuer.

4.1 Respondenter, företag och organisationer

Bank

Denna respondent är IT-chef för två bankkontor och ett mindre dotterbolag. Totalt sett har dessa tre företag ungefär 700 anställda, vilka alla påverkas av de beslut som fattas av IT-chefen. Respondenten har en bakgrund som civilingenjör, och har tidigare jobbat som konsult inom andra branscher såsom handel och industri. Han fungerar även som beställare mot den egna organisationen. Detta innebär att han styr budgeten och tar beslut på hög nivå för IT i företagen. Banken var en av de första i Sverige att införa "Internet-bank", och var även i frontlinjen med uttagsautomater, vilket tyder på en relativt stor tilltro till nya tekniska lösningar. Trots detta används idag ingen öppen programvara av företaget.

Produktutvecklingsföretag

Den andra intervjun är gjord på ett teknikinriktat produktutvecklingsföretag för utveckling av både hård- och mjukvara. Företaget anses av flera undersökningsföretag vara marknadsledande inom sin bransch och har kunder i nästan alla segment. Företaget är baserat i Sverige, men har även stor global spridning. En stor del av programvaruparken i företaget är utvecklad med öppen källkod. Personen vi intervjuat arbetar på IT-avdelningen på kontoret i Lund och har god insyn i de beslut som fattas avseende val av programvara. Respondenten har en bakgrund som matematiker men har tidigare arbetat med programutveckling.

Vårdenhet

Den tredje intervjun är gjord med en respondent från en vårdenhet i Skåne. Respondenten arbetar som IT-chef på den berörda enheten. Denna enhet är en av tolv stycken som finns i regionen. Inget av de program som används på enheten är öppen programvara. En central IT-ledning styr de tunga besluten, även över den berörda vårdenheten. IT-chefen har trots detta möjlighet att påverka beslut vid val av programvara. De beslut som IT-chefen tar, och medverkar till, berör cirka 850 personer vid enheten. Vidare har respondenten lång erfarenhet i branschen och har tidigare arbetat som IT-konsult i tio år.

Skola

Den fjärde intervjun är gjord på en skola som bedriver undervisning från förskola till årskurs nio. Skolan har ungefär 480 elever. Utöver detta verkar även 70 pedagoger samt ett mindre antal lokalvårdare och annan personal inom organisationen. Skolan har tidigare utfört tester med öppen programvara i organisationen. I dagsläget används öppen programvara till viss del inom verksamheten. Personen i fråga är anställd som IKT (Informations- och Kommunikationsteknik) -pedagog, och arbetar bland annat med att se över den programvara som används av både elever och personal på skolan. Trots att en del av besluten kring programvara ligger centralt styrt i organisationen har ändå respondenten viss inverkan vid dessa beslut.

4.2 Empiriskt resultat

Vi vill göra läsaren uppmärksam på att respondenterna i vissa frågor sätter den proprietära programvaran i relation till den öppna för att kunna svara på frågan. Som bekant jämför vi inte de båda alternativen, men eftersom den slutna programvaran som används idag täcker organisationens behov för tillfället, så fungerar den som en bra referenspunkt.

Citat som hämtats från transkriberingsprotokollen refereras enligt följande system: [X:Y]

X står för transkriptionsbilagans numrering, och Y står för den rad som citatet hämtats från. Exempelvis hänvisar [2:54] till intervju två, rad 54 i transkriptionsbilagan (Bilaga 4)

4.2.1 Respondenternas uppfattningar om funktionalitet

Bankens IT-chef anser att verksamhetens behov av funktioner i dagsläget även kan tillgodoses med öppen programvara. Även det resultat dessa funktioner uppnår anses vara: *"Lika bra eller lite sämre"* [1:178]. Däremot uppdagades under intervjun, negativa uppfattningar om öppen programvara vad gäller andra aspekter av funktionalitet. Exempelvis uttryckte respondenten en viss tvekan beträffande säkerheten med öppen programvara. Personen ansåg, att mer beprövad och etablerad programvara, dvs. i detta fall slutna programvara, är säkrare. Det betonas att detta är av yttersta vikt för den verksamhet som bedrivs inom banken. Vidare beskriver respondenten att öppen programvara kräver ett visst mått av egenutveckling för att få flera öppna programvaror att samverka. Detta anses inte vara praktiskt i detta företag då de uttryckligen *"[...]vill syssla med så lite egen utveckling som möjligt."* [1:204].

Respondenten på produktutvecklingsföretaget anser att de funktioner som krävs för att utföra arbetsuppgifterna inom företaget finns i öppen programvara. Respondenten anser också att den öppna programvaran har förmåga att återge samma resultat som med proprietär programvara. Öppen programvaras förmåga att interagera med annan programvara anses vara hög pga. det öppna filformatet som erbjuds. Med *"öppet format så kan jag då ta ner programmet och öppna direkt ju"* [2:148]. Det öppna filformatet anses således vara en av de anledningar som främjar samverkan mellan olika program. Den öppna programvarans säkerhet anses vara svår att bedöma då: *"problemet är ju att dom som slår, slår ju mest mot Microsoft-programvara för att det är det som ger störst utdelning."* [2:158]. Respondenten anser att det är svårt att bedöma säkerheten men känner sig trots allt säkrare med en öppen programvara.

IT-chefen på vårdenheten anser att de funktioner som krävs för att personal i organisationen skall kunna utföra sitt arbete, även tillgodoses med öppen programvara. Som exempel tas Microsoft Office, där det påpekas att alla funktioner i MS Office i dagsläget inte används. Funktionerna i det alternativ de övervägt, nämligen Open-Office, anses dock mindre, *"[...]men den är kanske fullt tillräcklig för det dagliga man använder"* [3:113]. Inom organisationen finns även behov av att flertalet program skall kunna samverka. Detta medför att, ifall ett program skall kunna bytas ut så måste det ersättande programmets output vara identiskt med det program som ersätts. Öppen programvara upplevs här vara något problematisk då man är osäker på just den förmågan. *"[...]samverkan kan ju vara mindre med öppen programvara."* [3:128]. Vad gäller säkerheten i öppen programvara anses den vara tillräckligt god för att motsvara de krav som ställs av organisationen på mjukvara. Detta beror

framförallt på att de använder en fristående brandvägg som är oberoende av typen av programvara.

De funktioner som behövs i organisationen anses av respondenten på skolan vara fullt tillräckliga i öppen programvara. Om dessa funktioner däremot genererar det resultat som eftersträvas är svårbedömt för intervjupersonen. När vi ställer frågan om öppen programvaras förmåga att samverka med andra program så berättar IKT-pedagogen om egna, faktiska upplevelser. Respondenten anser att det öppna filformatet som erbjuds av många öppna programvaror inbjuder till ökad integration med andra system. Vi exemplifierar med följande citat, "*Ja, vi har provat. Vi provade tidigare versioner av OpenOffice och det fungerade utan problem. [...] Jag har sett det ske så det är liksom inga problem.*" [4:76]. Säkerheten ses även kunna motsvara organisationens krav. Respondenten uttryckte att säkerheten var likvärdig i båden öppen och sluten program. Däremot "hackas" den öppna programvaran mindre. Intervjupersonen menar att detta har sin grund i att " *[...] dom som i största mån sysslar med att just hitta säkerhetshålen är mer positiva till den öppna programvaran, gör att den också hackas mindre.*" [4:80]

4.2.2 Respondenternas uppfattningar om tillförlitlighet

IT-chefen på banken har begränsad uppfattning om öppen programvaras mognad. Respondenten kan endast relatera till studier han läst och verkar inte själv ha reflekterat över frågan. Respondenten anses därmed inte ha en egen uppfattning om detta. Han svarade likadant om öppen programvaras återhämtningsförmåga och hade endast en vag uppfattning om detta. Vi kan dock spåra en misstro hos respondenten genom djupare analys av hela intervjun. IT-chefen nämnde bland annat att, " *[...] jag känner mig säkrare i en värld med mer beprövade produkter.*" och att när något väl havererar, så är det viktigt att de vet " *[...] hur man ska hantera ett fel och hur man går till väga och hur väl man känner till olika feltyper och åtgärder och vad förebyggande arbeten har sagt.*" [1:240]. Vidare menar respondenten att det inte finns lika mycket utarbetade åtgärder för olika feltyper " *[...] med öppen källkod där det är mindre beprövat*" [1:243]. Respondenten anser att det finns mindre resurser i form av yttre kompetens när systemet väl har kraschat, om man " *har valt ett smalare spår*" (=öppen programvara) [1:228]. Det anses således ta mycket tid att återställa öppen programvara, efter en krasch.

På produktutvecklingsföretaget anser respondenten att öppen programvara tidigare haft lägre frekvens av avbrott, pga. fel i dess konstruktion. Dock anser respondenten att sluten programvara kommit ikapp på senare tid. Respondenten menar att programvaran i sig självt spelar mindre roll här. Istället påpekas betydelsen av organisationen, dvs. förmågan att sätta upp programvaran på rätt sätt. Vad gäller programvaras förmåga att fortsätta fungera trots uppstått fel, så menar respondenten att det inte är någon större skillnad mellan öppen och sluten programvara. " *[...] det är i vart fall inte något minus för öppen programvara.*" [2:179]. Däremot har respondenten svårt att uttala sig om det skulle krävas större eller mindre ansträngning att återställa ett kraschat system. " *Det är svårt att generalisera...alltså när det går...det kan ha gått snett på så många olika sätt.*" [2:181]

På vårdenheten förklarade IT-chefen att han har mindre erfarenhet när det gäller tillförlitlighet vid öppen programvara. Trots detta har han uppfattningen att risken för avbrott är större i öppen programvara. Däremot är respondenten positiv till öppen programvaras förmåga att återhämta sig efter fel. Vidare påpekar respondenten att när applikationen väl kraschar

ordentligt så går det ”[...]nog åt mer resurser i öppen programvara, det är väl det som är lite problemet ju, det går nog mer åt resurser än slutna programvara” [3:162]

På skolan upplever respondenten ingen skillnad i avbrottsfrekvens mellan öppen och slutna programvara. Vid en eventuell uppkomst av ett oväntat fel, så upplever respondenten, att programvara beter sig på samma sätt, oavsett om den är öppen eller slutna. Vad gäller återställning av ett kraschat system berättar respondenten att de idag inte har något större system med öppen programvara. Dock upplevs det inte som att det skulle innebära större problem ifall sådant användes; inte ens för de mest affärskritiska systemen.

4.2.3 Respondenternas uppfattningar om användbarhet

IT-chefen på banken ansåg att det krävs större ansträngning för att förstå den logiska uppbyggnaden i öppen programvara jämfört med slutna programvara. Med andra ord så säger respondenten att ”[...]min uppfattning är nog ändå att de här mer större kommersiella produkterna dom, dom är lättare att använda.” [1:299]. Även exekvering av funktioner i öppen programvara anses vara svårare. Samtidigt menar intervjupersonen att öppen programvara är lätt att lära, om användaren inte har tidigare kunskap om hur den slutna programvaran används. "Om man står på noll i båda fallen tror jag det skulle vara lika lätt eller svårt [...] Men det är ju så att man har ju så mycket annat med sig, troligtvis, som gör att man inte står på noll" [1:302, 1:305]. Eftersom användare, enligt respondenten, inte står på noll, tolkar vi dennes svar så att det faktiskt är svårare att lära sig använda öppen programvara. När det kommer till applikationernas förmåga att tilltala användarna så anser han att den kan liknas vid slutna applikationers attraktivitet.

Respondenten vid produktutvecklingsföretaget ansåg, att den logiska strukturen i den öppna programvara som övervägts är lätt nog att förstå. Respondenten påpekar att öppenhet bidrar till att det är lättare att överblicka system. "När det gäller öppen källkod så kan man gå in och se hur installerar man det rent konkret. Då ser man ju också -OK, där är en sån applikationsserver där, och där e den och den modulen och det funkar så liksom." [2:186]. Öppenhet bidrar även till att öppen programvara blir lättare att lära sig. Dock påpekas att det finns stora skillnader mellan olika varianter av öppen programvara. De alternativ man framförallt övervägt är större program; dessa anses vara mycket väl dokumenterade. God dokumentation och tillgänglighet underlättar lärande om hantering av programvara. Att utföra uppgifter i arbetet med slutna eller öppen programvara upplevs som lika lätt/svårt. Vad gäller attraktivitet i programvara ansågs möjligheterna att personalisera öppen programvara vara större. Däremot ansågs slutna programvara i standardutförande vara mer tilltalande för användaren.

IT-chefen på värdenheten hade svårt att dra generella slutsatser angående öppen programvara och användbarhet. När vår fråga konkretiserades nämner respondenten att ”jag tror inte att det är bättre struktur i öppen programvara på det sättet, näe det tror jag inte. Det kan nog variera.” [3:203]. Svaret är svårt att tyda även om det kan tyckas vara i favör för den proprietära programvaran. Avsaknaden av standardiserade användargränssnitt innebär att den öppna programvaran är svårare att lära sig. Själva operationskontrollen, eller den ansträngning som krävs för exekvering av program och funktioner ses också som problematisk i öppen programvara. Möjligheten att modifiera och bygga om i användargränssnittet i den öppna programvaran ses dock som positiv, "i öppen programvara, där bygger du ju upp precis vad du vill ha [...] Så att det kan ju vara attraktivt ju.” [3:229]. Denna egenskap hos den öppna programvaran är tilltalande för användaren.

IKT-pedagogen ansåg att ansträngningen som krävs för att förstå den logiska strukturen i en programvara inte är olika stor beroende på om den är öppen eller sluten. De öppna programvaror som övervägts är väldigt lika sin slutna motsvarighet vilket underlättat förståelsen. *"Man har hittat en standard som folk har lärt sig, alltså såhär bör en webbläsare se ut."* [4:95]. Den ansträngning som krävs för att användaren skall lära sig tillämpa programvara på rätt sätt, upplevs som fullt acceptabel i de öppna programvaror man provat hittills. Detta trots att respondenten uttryckt att *"För dom lärarna som inte är datavana så är det helt klart så att det räcker med att flytta en knapp[...]för att det ska bli problem, och strula till sig."* [4:74]. Att utföra en specifik uppgift i öppen programvara upplevs som tillräckligt enkelt för att det skall gå att använda på denna skola. Öppen programvara upplevs vara mer tilltalande för användaren på grund av att *"[...]den öppna programvaran oftast ger möjligheten att anpassa till hur du själv vill ha det, till större del."* [4:103].

4.2.4 Respondenternas uppfattningar om produktivitet

Bankens IT-chef anser att öppen programvara kan leverera resultat snabbt. Respondenten nämner att *"[...]öppen programvara, skulle säkert, klara, klara bättre än andra som kommersiella produkter, med svarstid, och ehh, driftstabilitet, om det är ett begrepp."* [1:323]. IT-chefen tror inte att det krävs någon större mängd fysiska resurser för utförandet av funktioner i öppen programvara. Respondenten nämner istället att *"[...]det säkerligen är så att det kräver mer kapacitet i form av hårdvara också"* [1:329], när denne talar om sluten programvara i relation till öppen.

Respondenten på produktutvecklingsföretaget menar att det är väldigt svårt att uttala sig om generella skillnader, mellan olika typer av programvara med avseende på svars- och bearbetningstider för exekvering av funktioner. Viss skillnad upplevs mellan exempelvis klient- och server-programvara. Öppen programvara för klienter anses vara snabbare medan öppna serverprogramvaror upplevs långsammare. Skillnaderna mellan programtyper, blir dock allt mindre med tiden, enligt respondenten. Vad gäller resurser som krävs för att köra öppen programvara kontra sluten, anses det inte vara några större skillnader. *"[...]hårdvara det är rätt så hugget som stucket", "Personal e ju inte heller nån skillnad skulle jag vilja säg."* [2:213].

IT-chefen på vårdenheten ställer sig positiv till de snabba svars och bearbetningstider den öppna programvaran genererar. *"Ja det kan bli effektivare med [...] öppen programvara för då gör du bara det du precis behöver göra ju... Jag kan tänka mig det kan vara effektivare"* [3:235]. Vidare tror respondenten att det *"inte är någon större skillnad"* [3:237] i resursåtgång, från dagsläget (idag kör man endast sluten programvara), och hur det skulle se ut med en plattform på öppen programvara.

Respondenten på skolan har ingen uppfattning om svars- och bearbetningstider för exekvering av funktioner i öppen programvara i relation till organisationens behov. Vad gäller resurser som krävs för drift upplevs inte öppen programvara kräva allt för mycket hårdvara. Uppfattning saknas dock ifråga om andra resurser (i form av exempelvis personal och material) kräver mer än vad organisationen mäktar med.

4.2.5 Respondenternas uppfattningar om underhållsmässighet

Bankens IT-chef har litet intresse av att underhålla en programvara. Även om det är aktuellt med ett visst underhåll, så görs det inte i någon större utsträckning. Han menar att den öppna

programvaran erbjuder stora möjligheter till felsökning. Att modifiera programvara är även enklare i öppen programvara. Att validera den programvara som skall implementeras är också viktigt, då säkerhet och tillförlitlighet är något som prioriteras högt i organisationen. Vidare anser respondenten, att möjligheterna att testa öppen programvara är begränsade och inte motsvarar organisationens krav. Han menar att de gärna vill se någon annan testa programvaran innan de själva gör det. När en leverantör saknas, som enligt respondenten kan vara fallet med öppen programvara, är det svårt att hitta andra banker som använder öppen programvara. Möjligheten att kunna modifiera en del av programvaran, utan att en annan del av systemet blir påverkat är mindre i öppen programvara, enligt IT-chefen.

Att identifiera och analysera felkällor anses av respondenten på produktutvecklingsföretaget som ett mycket komplext arbete, oavsett vilken programvara det gäller. Det påpekas att öppenhet i källkoden ger möjlighet till noggrannare studier av öppen programvara, om så önskas. Ytterligare en konsekvens av öppen källkod är att det blir lättare att modifiera programvaran. Valideringen av modifierad programvara upplevs varken svårare eller lättare. Detta gäller även risken för att modifieringar skall få oönskade effekter.

Vårdenhetens IT-chef nämner att *"jag menar har du en sluten programvara så har du kanske, från leverantören, en vägledning, vad... som symptomen av fel och då kan man lättare ringa in felet"* [3:243]. Han hävdar därmed att öppen programvara kräver större ansträngning för identifiering av fel. Detta eftersom det är upp till användaren själv att göra det då, leverantörer saknas. Möjligheten att kunna modifiera öppen programvara ses som lättare i relation till organisationens nuvarande alternativ, den slutna programvaran. *"Ja där är ju också en fördel med öppen programvara, du kan modifiera de mer ju, än sluten programvara. Där måste du ju ta hänsyn till alla de andra intressenter som kör den programvaran ju."* [3:250] Återigen talar respondenten om avsaknaden av en direktleverantör för öppen programvara, när vi frågar om möjligheterna att kunna validera en modifierad applikation. Intervjupersonen svarar: *"[...]om man har öppen programvara det är ju det att man måste nog lägga ned lite mer tid på själva testandet"* [3:268]. Han påpekar att leverantörer av proprietär programvara sköter testandet åt dem. Respondenten anser att modifierationer i öppen programvara kan medföra oönskade effekter relativt ofta.

IKT-pedagogen på skolan påpekar att han personligen inte medverkat vid analys av brister och felkällor i den programvara som används inom organisationen. På grund av detta har han ingen uppfattning om detta arbete. Att modifiera öppen programvara upplevs som enkelt för den enskilde användaren. Organisationen har med både öppen och sluten programvara, upplevt problem att distribuera modifierad programvara till samtliga klienter. Respondenten har heller ingen insikt i de möjligheter öppen programvara ger i fråga om att validera modifierad programvara. Man anser att öppen programvara riskerar att påverkas på oväntade sätt av uppdateringar/uppgärningar. Ingen uppfattning finns dock, ifall risken är större eller mindre beroende på programvarutyp. Skillnaden mellan öppen och sluten programvara anses vara tillvägagångssättet för att ta del av information angående oönskade effekter. Med sluten programvara fås denna information främst via leverantörens support. Vid öppen programvara får man däremot själv ge sig ut på nätet och ta del av andra användares kommentarer och rekommendationer. Informationen finns alltså i båda fallen, men nås på olika sätt.

4.2.6 Respondenternas uppfattningar om flyttbarhet

Bankens IT-chef svarar på frågan om öppen programvara är tillräckligt lätt att anpassa till den egna verksamheten: "*kan den säkert vara i vissa fall, kan den nog vara.*". [1:455] Vidare förklarar respondenten att många gånger behöver de specialdesignad programvara som endast ska utföra en unik uppgift. Då skulle den öppna programvaran vara anpassningsbar och duglig för ändamålet. När det gäller den ansträngning som krävs för att installera öppen programvara fann vi en klar uppfattning om dess problematik. Respondenten nämner bland annat: "*[...]Jom man väljer det här programmet som har öppen källkod, att mycket av ansvaret hamnar liksom på oss [...] det vill vi helst inte ta, vi vill ju helst ha en leverantör som tar det ansvaret.*" [1:487]. Det handlar således igen om uppfattat avsaknad av en faktisk leverantör som kan ta ansvaret, och som banken kan falla tillbaka på. Installationsmomenten upplevs därmed som komplicerat i öppen programvara. Vidare tror respondenten att "*Det skulle vara väldigt resurskrävande.*" [1:373] att ersätta den befintliga plattformen med öppen programvara. Detta verkar gälla oavsett vilken programvara som skulle ersätta den som idag används i företagen.

Produktutvecklingsföretagets respondent menar att modifierbarheten i öppen programvara har ytterligare fördelar; det underlättar anpassning av program till andra miljöer. Sluten programvara upplevs ha begränsade möjligheter. Vad gäller installationsförfarandet, anses öppen programvara vara relativt lätthanterlig, "*[...]det är ju i vart fall inte svårare i allmänhet, det är faktiskt bättre.*" [2:243]. Vad gäller öppen programvaras förmåga att ersätta ett befintligt system inom organisationen, anses detta vara ett relativt stort projekt. Dock hävdas att ett dylikt projekt skulle vara fullt möjligt att planera och genomföra. Delar av ett det arbetet hade varit till nytta för företaget. Exempelvis skulle affärskritiska tjänster, som är långt bakåt kopplade i systemen, kunna identifieras.

IT-chefen för vårdenheten hävdar att öppen programvara är lättare att anpassa efter verksamheten. När vi frågar om hur ansträngande installationsmomentet för öppen programvara upplevs, så redogör respondenten för hur programvaran installeras på en server. Därifrån sprids den vidare ut till klienterna. Ansträngningen skulle således, enligt respondenten, inte påverkas ifall man installerar öppen eller sluten programvara. Vidare så säger intervjupersonen att det skulle vara ett omfattande arbete, även om det är möjligt, att byta från den befintliga programvaran till en öppen miljö.

IKT-pedagogen hävdar att anpassning av öppen programvara upplevs som tillräckligt lätthanterligt för organisationen. Detta grundas på erfarenheter från öppen programvara de har sedan tidigare i organisationen. Det visade sig då att anpassning till verksamheten inte var något större problem. Installationsförfarandet anses vara enklare i öppen programvara. Framst på grund av att: "*[...] du drar ner det direkt från nätet och plockar fram det. Det är liksom inte mer med det.*" [4:131]. Ersättningsbarheten i öppen programvara upplevs dock vara begränsad på grund av utbildningsaspekten. Respondenten förklarar att personal och elever måste läras upp för att kunna hantera programvaran korrekt. Däremot pekar respondenten på att det är positivt att lära sig hantera flera olika typer av programvara och att inte låsa sig vid ett alternativ.

5. Analys - uppfattningar i relation till forskning

Detta kapitel redogör för de empiriskt insamlade och sammanställda resultaten, i relation till forskningens syn på mjukvarukvalitet. För att underlätta förståelsen av de resultat vi kommit fram till, avslutas varje delkapitel med en tabell. Dessa tabeller skall endast ses som en förenkling av den analys som gjorts. Tabellerna bör alltså endast användas för att få en överblick över olika delkapitel.

5.1 Funktionalitet i öppen programvara

Samtliga respondenter är eniga om, att de behov av funktioner som finns i respektive verksamhet kan tillgodoses av öppen programvara (=”ändamålsenlighet”). Dessa funktioners förmåga att återge efterfrågat resultat ifrågasätts i viss mån (=”noggrannhet”). Framförallt är det vårdenheten som upplever det problematiskt då många av deras befintliga system är nära knutna till varandra. Skulle ett av dessa program bytas ut är det av stor betydelse, att ersättande program levererar samma resultat som det program som användes innan. Egentligen är det en fråga om en kombination av ”noggrannhet” och ”interoperabilitet” (interoperabilitet behandlas senare i detta avsnitt). I den vetenskapliga litteratur vi studerat finns en avsaknad av studier som behandlar detta ämne på vardaglig användarnivå. Dock anser forskningen att ”ändamålmässigheten” är hög; även det resultat denna genererar (=”noggrannhet”) i domäner som serverapplikationer (se kap 3.3.1). Om detta även gäller mer vardagliga program, är oklart enligt Samoladas och Stamelos (se kap. 3.3.1). Av denna anledning förespråkar de fördjupad forskning inom området.

Respondenterna från skola och produktutveckling uppfattar samverkan mellan olika program, även kallat ”interoperabilitet”, som god i den öppna programvaran. De menar att det öppna filformatet bidrar till att olika program lättare kan samverka. Å andra sidan upplever respondenterna för vård och bankverksamheten, en mer begränsad ”interoperabilitet” i öppen programvara. Samoladas och Stamelos (se kap. 3.3.1) hävdar, precis som skolan och produktutvecklingsföretaget, att ”interoperabilitet” i öppen programvara är god, främst på grund av dess öppna filformat. Således skiljer sig respondenternas uppfattningar från den teori som sammanställts.

Hoepman och Jacobs teori om ”säkerhet” (se kap. 3.3.1) tycks kunna förklara de olika uppfattningarna från respondenterna. Av de empiriska resultaten framgår att vissa anser att säkerheten i öppen programvara är tillräcklig, medan vissa hävdar dess motsats. Banken talar om att säkerhet är en viktig aspekt, och att de inte vill vara först på någon punkt. Detta är förstås fullt förklarligt då Hoepman och Jacobs visar att säkerheten initialt är låg i öppen programvara. Samtidigt stöds övriga respondenters uppfattningar om god säkerhet då författarna (se kap. 3.3.1) också hävdar att säkerheten i slutändan, är högre i öppen programvara. Det förtroende som sätts till säkerheten i öppen programvara förklaras av att det sker färre attacker mot mjukvaran i öppen programvara. Vi finner dock inga vetenskapliga belegg för att det faktiskt är på detta vis.

Tabell 5-1 – ”Funktionalitet” i öppen programvara

	Bank	Produktutv.	Vårdenhet	Skola	Forskning
Ändamålsenlighet	=	=	=	=	0
Noggrannhet	=	=	-	0	0
Interoperabilitet	-	+	-	+	+
Säkerhet	-	=	=	=	+

+ : Hög kvalitet

- : Låg kvalitet

= : Tillräcklig kvalitet

0 : Ingen data/teori

5.2 Tillförlitlighet i öppen programvara

Respondenternas syn på ”mognad” i öppen programvara är varierande. Produktutvecklingsföretaget och skolan anser att frekvensen avbrott på grund av fel i programkonstruktionen håller åtminstone acceptabel kvalitet. Vårdenheten anser att öppen programvara tenderar att orsaka fler fel, medan banken inte uppger någon åsikt alls avseende denna aspekt. Forskningen å andra sidan anser att öppen programvara uppvisar en hög grad av ”mognad” (se kap. 3.3.2). Respondenternas uppfattning om ”mognad” stämmer således huvudsakligen överens med forskningens åsikter avseende mognad i öppen programvara.

Avseende ”feltolerans” uppger samtliga respondenter förutom banken, att öppen programvara är åtminstone likvärdig med motsvarande slutna programvaror. Enligt vår litteraturgranskning har forskningen ej granskat denna aspekt av mjukvarukvalitet i öppen programvara. Således kan inga jämförelser göras här.

Vid analys ser vi återigen en spridd uppfattning i respondenternas syn på öppen programvaras förmåga till ”återhämtning” efter en krasch. Banken och vårdenheten har, som flera gånger förut, liknande uppfattningar. De anser att öppen programvara är mer problematisk att återställa till full funktion efter en krasch. Detta anses bero på avsaknaden av utifrån tillgänglig kompetens. Skolan upplever inte problemet vare sig större eller mindre, beroende på vilken typ av programvara man använder. Produktutvecklingsföretaget hävdar, att det inte går att dra några generella slutsatser i denna fråga och avstår därför från att ge ett utlåtande. Inte heller denna aspekt av mjukvarukvalitet i öppen programvara har undersökts vetenskapligt, och därför görs ingen jämförelse här heller.

Tabell 5-2 – ”Tillförlitlighet” i öppen programvara

	Bank	Produktutv	Vårdenhet	Skola	Forskning
Mognad	0	+	-	=	+
Feltolerans	0	=	=	=	0
Återhämtning	-	0	-	=	0

+ : Hög kvalitet

- : Låg kvalitet

= : Tillräcklig kvalitet

0 : Ingen data/teori

5.3 Användbarhet i öppen programvara

Även när vi utreder användbarhet urskiljer sig samma gruppering av respondenter; produktutvecklingsföretag och skola för sig, bank och vårdenheten för sig. Den ansträngning som krävs för att exekvera en funktion eller utföra en uppgift (="körbarhet"), anses enligt Samoladas och Stamelos (se kap. 3.3.3) som likvärdig mellan programvarorna. Respondenterna från skolan och produktutvecklingsföretaget uppfattar "körbarheten" som tillräcklig, eller bättre. Resterande respondenter ser dock detta som mer problematiskt, och att öppen programvara därmed håller en förhållandevis lägre kvalitet. Detta i motsats till forskarnas resultat (se kap 3.3.3).

Både banken och vården ansåg att den logiska strukturen i programvaran (="begriplighet"), var alltför komplicerad för att enkelt kunna förstå. Skolan och produktutvecklingsföretaget ansåg däremot att begripligheten var fullt tillräcklig för deras behov. Således är det bara två av respondenternas åsikter som återfinns i forskningen som hävdar att begripligheten är bristfällig i öppen programvara (se kap. 3.3.3).

En liknande uppdelning noterades bland våra respondenter avseende "lärbarhet". Banken och vården ansåg att lärbarheten var alltför låg för att öppen programvara skulle passa deras verksamheter. Produktutvecklingsföretaget och skolan ansåg dock att detta attribut inte var av låg kvalitet. Produktutvecklingsföretaget ansåg till och med att lärbarheten är högre i öppen programvara. Den forskning som sammanställts hävdar att lärbarheten är acceptabel (se kap. 3.3.3), vilket alltså endast överrensstämmer med två av respondenterna.

"Attraktivitet" i öppen programvara uppfattas av samtliga respondenter som tillräcklig, eller till och med väldigt hög. En persons uppfattning om vad som är attraktivt eller ej, är förstås subjektivt. Vi har därav haft svårigheter att validera dessa uppfattningar och jämföra med forskningen. I intervjuerna är det också genomgående att den öppna programvarans dynamiska gränssnitt är grunden till den högt uppfattade kvalitén i attributet "attraktivitet".

Tabell 5-3 – "Användbarhet" i öppen programvara

	Bank	Produktutv.	Vårdenhet	Skola	Forskning
Begriplighet	-	=	-	=	-
Lärbarhet	-	+	-	=	=
Körbarhet	-	=	-	=	=
Attraktivitet	=	=	+	+	0

+ : Hög kvalitet

- : Låg kvalitet

= : Tillräcklig kvalitet

0 : Ingen data/teori

5.4 Produktivitet i öppen programvara

Ingen av respondenterna upplever öppen programvara som för långsam avseende svars- och bearbetningstider vid exekvering av funktioner (=”tidsbeteende”). Istället anser banken och vårdenheten, att öppen programvara faktiskt är snabbare än motsvarande slutna programvara. Produktutvecklingsföretaget menar att öppen programvara tidigare varit snabbare i klientprogramvara, medan slutna programvara fungerat snabbare i serverapplikationer. Respondenten påpekar dock att skillnaderna mellan öppen och slutna programvara på senare tid successivt minskat avseende denna aspekt. Skolan väljer att inte utlåta sig om detta på grund av bristande erfarenhet och kunskap.

Samtliga respondenter var av uppfattningen att de resurser som utnyttjas vid utförande av funktioner i öppen programvara inte är större än vad som krävs av motsvarande slutna programvara (=”resursutnyttjande”).

På grund av bristande teoretisk referensram på denna punkt, kan vi ej jämföra respondenternas uppfattningar med litteraturen.

Tabell 5-4 – ”Produktivitet” i öppen programvara

	Bank	Produktutv.	Vårdenhet	Skola	Forskning
Tidsbeteende	+	=	+	0	0
Resursutnyttjande	+	=	=	=	0

+ : Hög kvalitet

- : Låg kvalitet

= : Tillräcklig kvalitet

0 : Ingen data/teori

5.5 Underhållsmässighet i öppen programvara

Genom våra intervjuer har det framgått att det finns en misstro mot underhållsmässighet i öppen programvara, dock inte på samtliga punkter. Samtliga respondenter påpekar att möjligheterna är goda att modifiera öppen programvara (=”föränderlighet”), vilket framförallt är en konsekvens av att källkoden är tillgänglig. Detta är enligt Laplante (se kap. 3.3.5), bara en av anledningarna till god underhållsmässighet i öppen programvara. Den främsta orsaken anses vara att utvecklarna lagt mycket arbete på att också göra källkoden lättförståelig. Detta för att fler utvecklare skall kunna ta del av den, och på så sätt kunna hjälpa till i utvecklingen. Ingen av respondenterna nämner dock detta som ett argument.

Banken och produktutvecklingsföretaget påpekar att arbetet med analys och identifiering av felkällor, åtminstone är lika enkel, eller enklare, i öppen programvara (=”analyserbarhet”). Detta stämmer överrens med Samoladas och Stamelos (se kap. 3.3.5) som hävdar att det inte föreligger någon skillnad mellan öppen och slutna programvara i detta avseende. Vårdenheten ställer sig däremot mer kritisk till detta, främst på grund av att man upplever att man står utan hjälp från leverantör. Skolan har ingen uppfattning om detta attribut.

Det som främst upplevs som ett problem är möjligheten att validera, dvs. utförligt testa modifierad programvara (=”testbarhet”). Dessa åsikter framförs främst av banken och vårdenheten. Detta beror på att de upplever att all testning måste utföras inom den egna verksamheten. Det påpekas av Laplante et al (se kap. 3.3.5), att användarna i större

utsträckning förväntas arbeta med testning av öppen programvara. Med sluten programvara förlitar de sig på tillverkaren eller leverantören, att noggrant undersöka varje modifiering. Produktutvecklingsföretaget däremot anser att testbarheten i öppen programvara är jämförbar med sluten programvara. Skolan har ingen uppfattning om detta attribut.

Respondenterna från bank och vårdenheten anser att modifieringar i öppen programvara riskerar att påverka oönskade delar av programmet (=”stabilitet”). Denna kritik delas av forskarsamhället (se kap 3.3.5). Produktutvecklingsföretaget i samstämmighet med skolan anser däremot att stabiliteten är jämförbar mellan programvarorna.

Tabell 5-5 – ”Underhållsmässighet” i öppen programvara

	Bank	Produktutv.	Vårdenhet	Skola	Forskning
Analyserbarhet	+	+	-	0	0
Föränderlighet	+	+	+	+	+
Testbarhet	-	=	-	0	-
Stabilitet	-	=	-	=	-

+ : Hög kvalitet

- : Låg kvalitet

= : Tillräcklig kvalitet

0 : Ingen data/teori

5.6 Flyttbarhet i öppen programvara

Samtliga respondenter upplever öppen programvara som lätt att anpassa efter den egna verksamheten. Detta har sin grund i, uttryckt av representanten för produktutvecklingsföretaget, att modifierbarheten är hög. Därmed blir den lättare att anpassa efter företagets behov. Enligt Samoladas och Stamelos (se kap. 3.3.6) beror detta på tillgången till källkoden, vilket underlättar ”anpassbarheten”.

Tre av respondenterna upplever installationsförfarandet (=”installerbarhet”) som relativt enkelt i öppen programvara. Då flertalet använder central distribution av programvara anses detta fungera oavsett ifall det är öppen eller sluten programvara. Det är intressant att se, att både banken och vårdenheten använder sig av samma argumentation om ett distribuerat installationsförfarande, dock är deras uppfattningar om kvalitén skiljda. Bankens åsikter skiljer sig eftersom respondenten menar att det inte finns hjälp att tillgå med installation av öppen programvara och därmed blir belastningen på den egna organisationen alltför stor. Samoladas och Stamelos (se kap. 3.3.6) hävdar i motsats till respondenterna att installation av öppen programvara vanligtvis är mer komplicerat. Framst beror detta på att viss öppen programvara, kräver parameterstyrning och tekniskt kunnande vid installation.

Att ersätta en befintlig programvara i någon av de fyra verksamheter som studerats (=”ersättningsbarhet”), upplevs som ett omfattande arbete av samtliga respondenter. Det ses dock som fullt möjligt i tre av fyra verksamheter Ett sådant byte kan innebära positiva effekter enligt produktutvecklingsföretaget; exempelvis möjliggör det identifiering av affärskritiska tjänster. Avsaknad av vetenskapliga studier i detta attribut gör att vi ej har möjlighet att jämföra uppfattningarna med forskningsresultat (se kap.3.3.6).

Tabell 5-6 – "Flyttbarhet" i öppen programvara

	Bank	Produktutv.	Vårdenhet	Skola	Forskning
Anpassbarhet	=	+	+	=	+
Installerbarhet	-	+	=	+	-
Ersättningsbarhet	=	=	=	-	0

+ : Hög kvalitet

- : Låg kvalitet

= : Tillräcklig kvalitet

0 : Ingen data/teori

6. Slutsatser och diskussion

I föregående kapitel presenterades vårt resultat i form av en analys. Utifrån denna analys drar vi här slutsatser och diskuterar dessa. Vi för också en diskussion över vad som påverkat studien samt hur vi som författare hade kunnat påverka den till det bättre. Vi presenterar även förslag för fortsatt forskning inom området.

6.1 Frågeställning ett

Hur uppfattas mjukvarukvalitet i öppen programvara av personer som beslutar om val av programvara i företag och organisationer?

Ingen av respondenterna uppgav att de använt sig av ISO/IEC 9126 vid utvärdering av öppen programvara. Dock kände respondenterna igen de flesta begreppen i modellen, vilket ledde till att vi kunde få merparten av våra frågor besvarade.

Uppfattningarna om mjukvarukvalitet i öppen programvara varierade mellan respondenterna. Frågeställningens avsikt var dock att undersöka ifall utveckling med öppen källkod, generellt sett, resulterar i programvara som anses ha hög eller låg mjukvarukvalitet. Trots de relativt varierande uppfattningarna, har vi kunnat notera ett par attribut av mjukvarukvalitet där man hade eniga åsikter.

Två av respondenterna, produktutvecklingsföretaget och skolan, var generellt sett positiva till öppen programvara. Dessa två ansåg att öppen programvara har hög eller tillräcklig kvalitet på samtliga attribut. Skolan upplevde dock "ersättningsbarheten" som låg. De positiva uppfattningarna, bekräftas av Laplante et al (2007) som hävdar att mjukvarukvalitet i öppen programvara kan överglänsa sluten programvara.

De andra respondenterna, banken och vårdenheten, uppvisade den skepsis som Laplante et al (2007) hävdar finns mot mjukvarukvalitet i öppen programvara. Dessa två upplevde ungefär att hälften av underattributen var bristfälliga. Dessa respondenter var eniga i sin negativa uppfattning om följande attribut:

- Interoperabilitet
- Återhämtning
- Begriplighet
- Lärbarhet
- Körbarhet
- Testbarhet
- Stabilitet

I denna lista återfinns vi bland annat tre av fyra underattribut för "användbarhet". Detta huvudattribut ter sig därför mest utmärkande. Pressman (se kap. 3.2.3) skriver, om användbarheten är låg, spelar det ingen roll hur bra funktionaliteten än är, programmet är ändå dömt att misslyckas.

Ingen av respondenterna ansåg att kvalitén var bristfällig i öppen programvara vad gäller följande attribut:

- Ändamålsenlighet
- Feltolerans
- Attraktivitet
- Tidsbeteende
- Resursutnyttjande
- Föränderlighet
- Anpassbarhet

Den uppmärksamme läsaren har nu noterat avsaknaden av ett par attribut som inte presenterats. Detta har sin grund i att resultaten som framgår i kapitel fem, varit så divergerande att slutsatser ej kunnat dras. Med ett större material och urval är det möjligt att vi hade kunnat dra slutsatser även om dessa attribut. Studien är explorativ till sin karaktär, men trots detta får våra slutsatser ses som en indikation på att en efterträdande studie antagligen resulterat i ett liknande resultat.

Kunskapsbidraget som lämnats i och med svaret på denna frågeställning, är en kompletterande bild av den skepsis mot mjukvarukvalitet i öppen programvara som Laplante et al påtalat. Utöver detta har vi visat på ett flertal attribut av mjukvarukvalitet som anses uppnå hög nivå i öppen programvara. Dessa resultat är förvisso intressanta, men blir mer värdefulla när de belyses i samband med ”frågeställning två”. Nästkommande delkapitel avser därför redogöra för likheter och skillnader mellan dessa uppfattningar och forskningens syn på mjukvarukvalitet i öppen programvara. På så vis kan de attribut av mjukvarukvalitet som inte uppfattats på rätt sätt identifieras.

6.2 Frågeställning två

Vad finns det för likheter och skillnader mellan respondenternas uppfattningar och vad forskningen anser om mjukvarukvalitet i öppen programvara?

Under sammanställningen av forskning kring mjukvarukvalitet i öppen programvara som gjorts (se kap 3.3) noterades brist på forskning inom området. Endast elva av de totalt 20 underattribut som ISO/IEC 9126 specificerat, hade granskats i öppen programvara. På grund av respondenternas varierande uppfattningar är det svårt att få fram en allmän uppfattning och jämföra med vad forskningen anser om mjukvarukvalitet i öppen programvara. Det visade sig att respondenternas uppfattningar både instämde och bestred de åsikter som återfinns i vetenskaplig litteratur.

Som framgått från föregående avsnitt noterades en större skepticism i urvalsgruppen bestående av respondenter från bank och vårdenhet. En berättigad fråga är hur väl denna skepticism återspeglas i vetenskapliga kontexter. Svaret och resultaten från undersökningen är dock divergerande men vi har observerat två tydliga fördelningar. Respondenternas misstro visar:

- Samstämmighet med forskningen.
- Motstridiga resultat jämfört med forskningen.

I den första delen av resultatet noterar vi en enad skepticism bland bank och vårdenhet som överensstämmer med forskningen. Denna typ av resultat innefattas av följande attribut:

- Stabilitet
- Testbarhet
- Begriplighet

Vi kan med detta resultat verifiera tidigare forskning som pekat på de attribut som varit undermåliga. Med kunskap om dessa attribut torde det nu vara ännu tydligare för utvecklare av öppen programvara över vad som är och uppfattas som bristande. Vidare borde denna kunskap ge uppslag för vad som behöver ytterligare förbättringar för att nå ett eventuellt genombrott för kommersiellt användande.

I den andra delen av resultatet har vi noterat en skillnad mellan upplevd och vetenskapligt bevisad mjukvarukvalitet. Vi kan därför bekräfta den teori som varit aktuell för rapportens problemformulering. Laplante et al (2007) pekar på skillnader mellan uppfattningar och forskningens syn på öppen programvaras mjukvarukvalitet, men de nämner endast mjukvarukvalitet som ett helhetsbegrepp. Vi har däremot gjort en djupanalys och redogjort för varje attribut som utgör helhetsbegreppet. Därmed bekräftas inte bara denna teori utan kan också komplettera Laplante et als teori. Detta görs genom att peka på följande specifika attribut som upplevs annorlunda av den grupp som ställer sig skeptiska till öppen programvara:

- Interoperabilitet
- Lärbarhet
- Körbarhet

Till diskussionen hör även det faktum, att banken och vårdenheten båda representerar en verksamhet där öppen programvara inte används. Detta styrker Twaileh och Rana (2006) påstående som nämnts inledningsvis i denna rapport; de som inte är introducerade till öppen programvara nås aldrig av informationen om mjukvarans kvalitet. Att kvalitén uppfattas som bristande har grund i respondenternas okunskap enligt dessa författare. Samtidigt som vi bekräftar Twaileh och Rana's teori så kan vi ge en detaljerad fingervisning för utvecklare av öppen programvara, för exakt vilka attribut som felaktigt uppfattas som bristande. Vilka åtgärder som bör vidtas för de attribut som inte uppfattas "rättvist" med forskningen i åtanke, är upp till annan forskning att besvara.

Hittills har vi endast diskuterat och dragit slutsatser utifrån de respondenter som visat en skepticism antingen i enighet eller i motstridighet med vad forskare visat. Även de positiva uppfattningar vi har identifierat i öppen programvaras mjukvarukvalitet är av yttersta intresse. Som vi fastställt i föregående frågeställning (se kap 6.1) finns det ett flertal attribut som av samtliga respondenter uppfattas hålla en acceptabel eller till och med hög kvalitet. I fem av sju av dessa attribut saknar vi olyckligtvis vetenskapliga belegg för att kunna göra den jämförelse som studien bland annat syftar till. I de två fall där det ges möjlighet att skildra den enade empirin med tidigare forskningsresultat ser vi en god överensstämmelse. Detta talar för att de fem andra attributen också kan ses som representativa för den faktiska mjukvarukvalitén, något som dock måste verifieras med vetenskapliga studier. Det bör dock tas i beaktande att denna studie är av explorativ karaktär. Därmed bör våra slutsatser tas som en indikation på hur det kan vara. Med ett större urval hade det varit lättare att dra generella slutsatser.

Genom att presentera detta bidrag hyser vi förhoppningar att andra kan ta vid. Vi hyser även hopp om att denna kunskap inte endast tilltalar studenter och forskare utan även de som utvecklar öppen programvara. Med den kunskap som presenterats bör utvecklare kunna lyfta utvecklingsfokus från de attribut som visat sig vara fullt tillräckliga. Istället borde de arbeta för en bättre kvalitet i de attribut som visat sig vara undermåliga. Kunskapsbidraget torde även ge indikationer för utvecklandet av ett medium med möjlighet att förmedla öppen programvaras faktiska kvalitet till de som ännu inte är insatta i frågan.

Sammanfattningsvis hoppas vi att denna kunskap är ett steg på vägen för att stödja den revolution som nämns inledningsvis i rapporten.

6.3 Egna reflektioner

Uppsatsarbetet har varit både spännande och intressant. Valet av en kvalitativ forskningsmetodik har varit mycket lärorik och användbar. Hela processen med definition av frågor, kodning, kategorisering, insamling av data och presentation har dock varit mer tidskrävande och mödosam än vi kunde tänka oss. Därav följer förklaringen till det begränsade antalet respondenter. Det var också givande, att under intervjuerna få ta del av hur man resonerar kring val av programvara ute i företag och organisationer. Vi kan konstatera att respondenterna överlag upplevdes som nyfikna på öppen programvara och såg med intresse på dess framtid. Flertalet respondenter trodde också att öppen programvara kommer att användas i större utsträckning om ett par år.

Vi blev förvånade över den lilla uppmärksamhet mjukvarukvalitet i öppen programvara har fått i vetenskapliga sammanhang på detaljnivå. Detta faktum är viktigt att betona då det finns mycket skrivet om den generella mjukvarukvalitén. Vi har granskat mjukvarukvalitén på djupet och sökt vetenskapligt förankrade teorier som berör mjukvarukvalitén på attributnivå. Anledningen till att det saknas litteratur på denna nivå, kan bero på att öppen programvara, som ett alternativ för företag, trots allt är ett relativt nytt fenomen. Ett exempel på avsaknaden av forskning nämns i teorikapitlet (se kap. 3.3.4). Där har vi inte kunnat finna vetenskapligt belägg för något av underattributen av produktivitet i öppen programvara. Detta förvånar oss, då det rimligtvis borde vara det attribut som enklast går att mäta. Svarts- och bearbetningstider för exekvering, samt nödvändiga resurser för utförande av funktioner borde enkelt gå att mäta då det man kan tillämpa kvantitativ forskningsmetodik. Trots detta upplever vi en avsaknad av studier på denna punkt. Då vi som författare får betraktas som noviser inom kvalitativ forskningsmetodik, är det inte omöjligt att vår litteratursökning har brister. Detta kan innebära att vi omedvetet missat litteratur som behandlar de attribut vi påstått saknas i vetenskapen. Våra förhoppningar är dock att så inte är fallet, då vi försökt vara så noggranna som möjligt i den litteratursökning som gjorts som framgår av metodkapitlet (se kap 2.4.1).

När vi ser tillbaka på de intervjuer vi utfört, kunde vi ha varit mer tydliga i våra frågor för att undvika att svaren skulle bli svävande. Detta gjorde analyserna mödosamma. Det är en svår balansgång att ställa tillräckligt preciserade frågor för att inte missa viktig information, samtidigt som man ser till att inget onödigt diskuteras. Respondenten från skolan var den person som kunde svara på minst antal av våra frågor. Som framgår i analysen (se kap. 5), uteblev svar på totalt fyra av våra 20 frågor. Detta beror på att, en del av utvärderingen av programvara inom organisationen sköts av annan personal än respondenten själv. Tyvärr har vi inte kunnat komma i kontakt med dessa personer för att komplettera med information från skolan. Det är dock inte säkert att dessa personer skulle kunna besvara de resterande frågorna.

Det kan vara så, att de inom skolan inte granskat öppen programvara avseende de attribut vi ej funnit uppfattningar om.

Den kvalitetsmodell vi valde, ISO/IEC 9126 har ur ett undersökningsperspektiv varit problematisk att jobba med. År 1991 kom den första versionen av kvalitetsmodellen; tio år senare reviderades modellen. Av vår litteratursökning att döma, har det varit väldigt svårt att hitta detaljerad information om den reviderade modellen från 2001. Vi tog därför beslutet att använda oss av den äldre modellen, ISO/IEC 9126:1991. På grund av avsaknad av litteratur från den senare versionen är det svårt att uttala sig hur detta har påverkat vår studie. Det är däremot uppenbart att om vi använt den senaste revisionen så hade det påverkat studien i någon form. Detta eftersom vi identifierat minst en skillnad i modellen (se kap 3.1.2), även om den är liten. Även ISO/IEC 9126:1991 visade sig svårtillgänglig eftersom den kostar pengar att ta del av. Detta tvingade oss att använda sekundära källor för att beskriva modellen. För att återge modellen så korrekt som möjligt har tre olika källor använts. Ett attribut i modellen, (*conformance*), råder det viss oklarhet över. Någon enighet om detta attributs relevans i modellen, fanns inte bland de tre källor som använts. Vi har därför valt att utelämna detta attribut. Vi tror dock inte att detta fått några drastiska konsekvenser för resultatet i studien i sin helhet.

6.4 Förslag till fortsatt forskning

Vi har under arbetets gång uppmärksammat avsaknaden av vetenskapligt utförd granskning av mjukvarukvalitet i öppen programvara. Liksom Samoladas och Stamelos (2003), ser vi behovet av ytterligare forskning på området.

Eftersom vår studie är av explorativ karaktär ser vi gärna att någon tar vid där denna studie slutar. Förslagsvis kan specifika branscher studeras mer ingående. På så vis skulle det gå att konstatera om uppfattningarna är beroende av den bransch som studeras.

7. Referenser

- Abrahamsson, A. & Furu, H. (2004). *Faktorer för mjukvarukvalitet Vad är viktigt för slutanvändaren?* Luleå: Luleå Tekniska Universitet, Institutionen för Industriell ekonomi och samhällsvetenskap.
- Backman, J. (1998). *Rapporter och uppsatser*. Lund: Studentlitteratur.
- Bazzana, G., Andersen O., Jokela T. (1993). *ISO 9126 and ISO 9000: friends or foes?* Software Engineering Standards Symposium, 1993, 79-88.
- Blomquist, R. (2000). *Mått på kvalitet - Hur IT-företag bedömer kvalitet*. Göteborg: Göteborgs Universitet, Institutionen för Informatik
- Bryman, A. & Bell, E. (2003). *Företagsekonomiska forskningsmetoder*. Malmö: Liber
- Computer Sweden (2007). *CS kompendium om öppen källkod*. [Internet] Finns tillgänglig på: http://cstjanster.idg.se/globalincludes/applikationer/pdf_arkivet/moduler/sponsorsite.asp?SponsorSiteID=49 [besökt 30okt 2007]
- Côté, M., Suryn, W., Georgiadou, E. (2007). *In search for a widely applicable and accepted software quality model for software quality engineering*. Software Quality Journal, 15, (4), 401-416.
- Expowera (2007). *LOU, Lagen om offentlig upphandling*. [Internet] Finns tillgänglig på: http://www.expowera.com/mentor/juridik/juridik_lou.htm [besökt 22dec 2007]
- Fjellman, C., Neste, N. (2005). *Uppfattningar, tankar och känslor – organisationsidentitet på Migrationsverket*. Karlstad: Karlstad Universitet, Institutionen för ekonomi.
- Garvin, D.A (1984). *What Does "Product Quality" Really Mean?* Sloan Management Review, 26, (1), 25
- Hoepman, J., Bart, J. (2007). *Increased security through open source*. Association for Computing Machinery. Communications of the ACM, 50, (1), 79-83.
- Håkansson, B. (2000). *Egendeklaration av programvaror - svensk metod för kvalitetsmärkning av programvaror med internationella standarder som grund*. Stockholm: Nykopia Tryck AB
- Kitchenham, B & Pfleeger, S.L (1996). *Software Quality: The elusive target*. IEEE Software, 13, (1), 12-21.
- Landén, M. (2007): *Att söka vetenskapliga artiklar - En vägledning i systematisk litteratursökning* [Internet] Finns tillgänglig på: <http://www.mfb.medfak.lu.se/Informationssokning.pdf> [besökt 19nov 2007]

- Laplante, P., Gold, A., Costello, T. (2007). *Open Source Software: Is It Worth Converting?* IT Professional, 9, (4), 28-33.
- McCall, J., Richards, P., & Walters, G. (1977). *Factors in software quality*. Griffiths Air Force Base. New York: Rome Air Development Center Air Force Systems Command
- Nationalencycledin (2008). *Nationalencycledin* [Internet]
Finns tillgänglig på: <http://www.ne.se> [Besökt 2jan 2008]
- Nielsen, J. (1993). *Usability Engineering*. San Diego: Academic Press
- Ny Teknik (2007). *Så ska öppen källkod slå igenom* [Internet]
Finns tillgänglig på: <http://www.nyteknik.se/art/50809> [besökt 30okt 2007]
- Paulson, J.W., Giancarlo, S., Armin, E. (2004). *An Empirical Study of Open-Source and Closed-Source Software Products*. IEEE Transactions on Software Engineering, 30, (4), 246-256.
- Pressman, R. (2001). *Software Engineering A practitioner's approach*. Singapore: McGraw-Hill Book Co.
- Raghunathan, S., Prasad, A., Mishra, B.K., & Chang, H. (2005). *Open Source Versus Closed Source: Software Quality in Monopoly and Competitive Markets*. IEEE Transactions on Systems, Man and Cybernetics, Part A, 35, (6), 903-918
- Samoladas, I. & Stamelos, I. (2003). *Assessing Free/Open Source Software Quality*. Thessaloniki: Aristotle University of Thessaloniki, Department of Informatics.
- Statskontoret (2003): *Öppen Programvara* [Internet]
Finns tillgänglig på: <http://www.statskontoret.se/upload/publikationer/2003/200308.pdf>
[besökt 29okt 2007]
- Sveriges Radio (2003): *Staten kan spara miljarder på att byta dataprogram* [Internet]
Finns tillgänglig på: <http://www.sr.se/ekot/artikel.asp?artikel=182241> [besökt 29okt 2007]
- Tawileh, A. & Rana, O. (2006). *Free and Open Source Software Quality Assurance*. Information and Communication Technologies, 2006. ICTTA '06. 2nd, 2, 2866-2871
- Wallén, G. (1996): *Vetenskapsteori och forskningsmetodik* (2:a rev. uppl.). Lund: Studentlitteratur

8. Bilagor

8.1 Företagsbrev

Hej [Namn]

Vi är oerhört tacksamma för att du vill hjälpa till och delta i vår studie. Det ska bli väldigt spännande att få höra dina åsikter och uppfattningar kring öppen programvara.

Syftet med vår studie är att ge en ökad förståelse för hur öppen programvaras mjukvarukvalitet uppfattas av personer som beslutar om val av programvara i företag.

Vi är därför intresserade av att diskutera uppfattningar om mjukvarukvalitetsfaktorer kring öppen programvara. Dessa faktorer innefattar bland annat:

- * Användbarhet
- * Funktionalitet
- * etc.

Samtidigt vill vi passa på och upplysa dig om att både din egen och er organisations anonymitet kommer att garanteras i den rapport som läggs fram för allmänheten (såvida annat inte önskas). Intervjun kommer att spelas in på ljudband, och de enda personer (förutom vi), som eventuellt kommer att lyssna på bandet är de examinatorer som bedömer kandidatuppsatsen.

Återigen, ett stort tack för att du ställer upp!

Med vänliga hälsningar

Jonas Persson och Jakob Mattelin

[Telefonnummer] [Telefonnummer]

8.2 Respondentvaliderings brev

Hej [Namn]

Efter ”mycket möda och stort besvär” så är vi klara med en preliminär analys av det material som vi samlat in från intervjun med dig.

Vi skickar med den tolkning vi gjort av dina uppfattningar. Är det så att du tycker att vi inte tolkat vad du framfört på ett korrekt sätt, skulle vi vara tacksamma om du berättade det för oss så att vi kan rätta till eventuella fel.

För övrigt är vi återigen jätteglada för att du ställde upp och önskar dig en God Jul och ett Gott nytt år!

Jonas och Jakob!

Tolkning av [Företagsnamn]

[Här bifogas den tolkning vi gjort av respondentens svar, hämtat från kap 4.]

8.3 Intervjuguide

Respondenten ombeds tänka sig in i och kortfattat beskriva situationer, där han/hon varit med och beslutat om val av programvara, där öppen programvara på ett eller annat sätt varit ett av de tillgängliga alternativen. Vi vill inte veta vilka krav som var viktiga vid valet, utan snarare vad personen upplevde att öppen programvara hade att erbjuda och om detta överrensstämmer med respondentens krav eller ej. Om det eventuellt skulle vara så att respondenten inte har någon uppfattning så ombeds denne att framföra det.

1. Hur väl anser du att funktionerna i öppen programvara motsvarar behoven i er organisation?
2. Hur upplever du att det öppna programvaru alternativet uppnår det resultat ni eftersträvar ?
3. Hur upplever du öppen programvaras förmåga att samverka med era andra program och system?
4. Vad anser du om säkerheten i öppen programvara (att obehöriga inte får tillgång till data?)
5. Har du någon uppfattning om antalet oväntade avbrott pga. fel i den öppna programvaran?
6. Hur upplever du att programmets förmåga, att trots fel, kunna fortsätta fungera?
7. Vad anser du om öppen programvaras förmåga att hantera en krasch, dvs. hur mycket resurser i form av tid och kraft som krävs för att återställa ?
8. Vad anser du om enkelheten/svårigheten att förstå uppbyggnaden av programmets struktur i öppen programvara?
9. Vad är din uppfattning om hur lätt/svårt det är att lära sig använda öppen programvara rätt?
10. Anser du att det är tillräckligt lätt att utföra en uppgift i den öppna programvaran?

11. Vad anser du om programvarans förmåga att tilltala användaren (attraktivitet)?
12. Hur upplever du svars- och bearbetningstider för exekvering av funktioner i öppen programvara?
13. Vad är din uppfattning om hur mycket resurser öppen programvara kräver? (ex: hårdvara, mjukvara, personal och material)
14. Vad är din uppfattning om det arbete som krävs för att analysera eller identifiera brister och felkällor?
15. Hur stora möjligheter anser du att det finns att modifiera/uppgradera programvaran?
16. Har du reflekterat hur det är att i så fall testa den modifierade programvaran?
17. Upplever du att uppdateringar även kan ha oönskade effekter på programvaran?
18. Tycker du att öppna programalternativ har varit lätta att anpassa till er verksamhet?
19. Hur upplever du de ansträngningar som behövs göras för installation av öppen programvara?
20. Hur omfattande anser du att ett byte av er befintliga programvara, till öppen programvara skulle vara/ har varit?

8.4 Transkriptionsbilagor

8.4.1 Bank

1	Jakob: Vi spelar in detta nu och sen så går vi hem i veckan och transkriberar, skriver ut i text.
2	Bank: Ja
3	Jakob: och sen så, kommer vi väl att maila det till dig så kan du väll titta igenom det och se så att det stämmer, det gör du som du vill, liksom vi vill inte hitta på något.
4	Jakob: Men om du vill så bara kolla igenom och godkänn, så, kommer vi sen att använda det.
5	Bank: Mm
6	Jonas: Vi kommer ju att göra någonting, någon slags tolkning av transkriptionen
7	Jakob: Mm
8	Jonas: Så att vi kan se att vi tolkat dina svar rätt, så att du inte tycker att vi är helt fel ute
9	Bank: Nä
10	Jakob: Det beror lite på svaren här, ifall det blir väldigt långa så kanske vi tar kontentan
11	Jonas: Mm, Jaa
12	Jakob: så att tolkningen går rätt till
13	Bank: Jag förstår, jag har också skrivit ett examens arbete en gång i tiden
14	Jakob, Jonas: Ja
15	Jakob: Då är du också insatt i transkriberingsarbetet
16	Bank: Ja ungefär
17	Jakob: Ja det är inte världens mest stimulerande kanske
18	Bank: Men ni får gärna börja och berätta lite om bakgrunden och varför ni göra detta:
19	Jakob: Mm
20	Bank: Det skulle vara kul att veta
21	Jakob: Vi tänkte väl, det hela började med att vi snubblade över egentligen en artikel som sa att det här med opensource. Vi kan kalla det öppen programvara.
22	Bank: Ja
23	Jakob: För enkelheten skull, ehh, alltså det är ett fenomen som egentligen, som många hävdar motsvara de behov som finns i organisationer och företag.
24	Bank: Ja
25	Jakob: Och oftast är det lägre kostnader att använda sig av öppen programvara men ändå är det väldigt få som faktiskt använder det.
26	Bank: Mm
27	Jakob: Och det, det känns lite motsägelsefullt, för att oftast så att finns kostnadsbesparingar, det är ju någonting som oftast eftersträvas av många företag.

28	Bank: Mm
29	Jakob: Och trots det är det ingen som använder det, så tänkte vi väll kolla vad det kan bero på. Och en teori som vi har hittat är att det är mjukvarukvalitet som påverkar detta beslutet.
30	Bank: Mm
31	Jakob: Så nu tänkte vi gå runt på ett par organisationer och, ehh, höra oss för uppfattningar om mjukvarukvalitet
32	Jonas: Mm
33	Jakob: Det är väl de egenskaper i programvaran, som, ja. Så dom här personerna som vi tänkte intervjua är väl personer som antingen är delaktiga i eller i alla fall har en insikt i den beslutsprocess kan man väl säga där man tar beslut om vilken programvara som skall användas.
34	Bank: Mm
35	Jakob: Och höra deras uppfattningar är.
36	Jonas: Mm, och de som har insikt i det valet.
37	Bank: Ja, just det
38	Jonas: Sen finns det teori som säger att mjukvarukvalitet, viss teori säger att den är bättre i opensource, viss säger att den är sämre. Så därför vill vi gå ut och titta hur det är i praktiken.
39	Bank: Man kan ju då säga vad är då kvalitet
40	Jakob: Mm, precis ja, så det blir, ja, absolut, ehh. Förstår du då lite idén.
41	Bank: Ja då förstår jag.
42	Jakob: Ja
43	Bank: Vad ni är ute efter.
44	Jakob, Jonas: Ja, Mm
45	Jonas: Ehh, jag vet inte om du har något, ehh, konkret, någon konkret, någon konkret situation
46	och sätta dig in i och såhär, ehh.
47	Bank: Jaa
48	Jonas: och när
49	Bank: Ja men ja näe, såsom
50	Jonas: ehh
51	Bank: så som jag tänker kring det hela, ehh för rent typ att det är en budget process
52	Jonas: Mm
53	Bank: Ehh, och budget och planering för nästa år och några år framåt och så, ehh. Vi börjar titta över våra strategier inom det här området, hur det ska se ut på längre sikt, det är klart, att då behöver man välja, och ska dra mot varandra, och bestämma, antingen går vi ditåt eller ditåt, eller så
54	Jakob: Mm
55	Bank: Ehh och det är klart det finns ju några bitar här, direkt när jag tänker opensource då, jag vet inte, jag har bara förknippat det mer med sådana här, klient program som utför olika saker.
56	Jonas: Mm
57	Bank: Till exempel så finns det väl, jag vet, när jag läste så fanns det en ersättare till Office paketet ju.

58	Jonas, Jakob: Mm
59	Bank: Star-Office eller vad det heter
60	Jonas, Jakob: Mm
61	Bank: som var gratis
62	Jonas: Mm
63	Bank: Och sen var det olika databaser och så vidare, det finns ju också, det kostar ju ganska mycket, vi betalar ju stora summor för licensen till databaserna, ehh.
64	Jonas: Mm
65	Bank: Ehh, men jag vet ju också att det finns andra varianter nu som är betydligt billigare, men egentligen när jag tänker såhär, som ni sa med kvalité här, ehh tidigare, så är det mycket, många aspekter i det. I de här programmen som Office paketet, eller liknande program eller så som användare kommer i kontakt med.
66	Jakob, Jonas: Mm
67	Bank: Och dels är det ju så att en aspekt i kvalité är ju såklart att det ska fungera, det här programmet.
68	Jonas, Jakob: Mm
69	Bank: Ehh, men sen så måste det ju också vara enkelt, för användaren för det är ju den typ av program som alla använder ute och ska använda.
70	Jonas: Mm
71	Bank: Ehh, vi har ju dessutom en, en intern helpdesk som hjälper till vid alla datorrelaterade problem, och där märker man ju ganska snabbt om vi gör något. Om vi tar ett beslut och sjesätter någonting som då inte uppfattas som enkelt kanske, så får du ju massor att göra ändå.
72	Jakob: Ja
73	Bank: Det är ju ganska stor kostnad att ta hand om allt det där.
74	Jonas: Mm
75	Bank: Ehh, lika dant så får det här ju inte så mycket den effekten som vi vill hos de som ska använda det. De kanske nyttjar det till en liten del och sen så gör de resten på papper precis som de gjorde tidigare.
76	Jonas, Jakob: Mm
77	Bank: Då har vi kastat pengar i sjön ju.
78	Jonas, Jakob: Mm
79	Bank: Ehh och sen, ehh ja, allt som uppfattas som enkelt är väl ofta, jag menar, ja vad är enkelt? Ja till viss del är det ju såklart hur mycket utbildning man lägger på det där som ska användas. Och sådant där dagligt använt som Office paketet till exempel igen då, då, så får jag för mig att åtminstone att dom allra flesta som har läst någon slags datorutbildning, det har ju på något sätt lärt sig Officepaketet eller använder det i alla fall.
80	Jonas: Mm precis.
81	Bank: Och med licenskostnaden i sig är mycket, mycket högre med Office paketet, och det är en ganska stor kostnad i sig då, för en sådan här organisation med ett par hundra anställda och vi ska köpa licenser till alla
82	Jonas: Mm
83	Bank: Så det är ändå, om man tittar total kostnadsmässigt så är det, då finns det ju andra aspekter i det hela, det är klart att det är det här med, vad får vi ut av det i andra ändan
84	Jonas: Mm
85	Bank: Kan folk nyttja det när vi kommunicerar med omvärlden, kan de ta emot de här filerna som vi skickar, uppfattas vi som enkla ut mot kunder och andra.

86	Jonas: Mm
87	Bank: Ehh, jag menar sen hur mycket frågor, hur mycket utbildning måste vi lägga på internt här, hur mycket kan vi förmoda att användaren själv kan.
88	Jonas: Mm
89	Bank: Då kommer den vi anställt här, och så vidare, det är klart, det är en kostnad för oss också, vi måste utbilda kontinuerligt.
90	Jonas, Jakob: Mm
91	Bank: Och sen, ehh, det är klart när vi inte lyckas med utbildningen riktigt och det kommer upp frågor vid helpdesken, det är en kostnad.
92	Jonas, Jakob: Mm
93	Bank: Det är klart, det är olika felhanteringar och så vidare.
94	Jakob: Mm
95	Bank: Det är egentligen dom här bitarna runt omkring som, många om mycket gör det, lika dant är det hos dom här teknikerna som ska ta hand om de här frågorna, om vi blir för spretiga, och väljer sånt som dom inte är duktiga på, det är klart att då blir det svårare för dom att hjälpa dom här vanliga användarna.
96	Jonas: Mm
97	Bank: Till exempel bara när det gäller Office paketet så skulle det varit väldigt, väldigt, väldigt mycket som skulle, alltså, ha förutsättningar som skulle behöva ändras för att vi skulle välja något annat egentligen trots att licens summan är en stor kostnad i sig, av alla dom aspekterna
98	Jakob: Det är intressant det du säger, det är väl lite såhär vi hade tänkt lägga upp intervjufrågorna, att vi utgår från t.ex. funktionalitet och användbarhet och sådär, så har vi delat upp det i lite mindre punkter och så som vi tänkte gå igenom.
99	Bank: Ja
100	Jakob: Lite och se, tänkte bara höra innan vi bränner av ett par frågor här.
101	Bank: Ja
102	Jakob: Om du kan beskriva din tjänst här lite gran, på [företagsnamn].
103	Bank: Vad jag gör för någonting, och så?
104	Jakob: Mm
105	Bank: Ja, jag jobbar som IT-chef
106	Jakob: Mm
107	Bank: för den här banken och sen så jobbar vi väldigt tigt ihop med [företagsnamn], jag har även den rollen på den banken, och sen så äger vi gemensamt ett dotterbolag som heter [företagsnamn], som är egentligen, ehh ja inom mitt område så utgör den då våran IT avdelning kan man säga. Vi har sett att vi har så lika verksamhet med våran bank som vi jobbar tigt ihop.
108	Jonas: Mm
109	Bank: Så vi gör det tillsammans där istället då
110	Jonas: Ja, okej
111	Bank: för att kunna pressa kostnader på det sättet
112	Jakob: Mm
113	Bank: Som till exempel helpdesk och drift och förvaltning av de olika systemen ligger där. Och tidigare har vi haft viss egen utveckling som vi gjort där. Så min roll är egentligen att, jag, jag är som en beställare mot vår egen

	organisation. Man kan säga att jag styr budgeten och sen tar jag beslut på ganska hög nivå.
114	Jonas: Mm okej
115	Bank: Ehh men sen när det gäller olika delar och så, så försöker vi göra att de här [företagsnamn] är så självständiga som möjligt så att det tar besluten inom sina områden, men klart, strategin måste vi ju, måste vi ju jobba efter samma strategi.
116	Jakob: Mm
117	Bank: Ehh, kan man väl säga
118	Jakob: Mm
119	Jakob: Vad har du för bakgrund egentligen innan du hamnade här.
120	Bank: Ja, akademiskt så har jag läst till civilingenjör, ehh, och har jobbat inom olika konsultverksamheter, egentligen inte helt andra branscher än finansbranschen, jag har jobbat inom industri och handel framförallt.
121	Jakob: Mm
122	Bank: Levererat affärskritiska system
123	Jakob: Mm
124	Bank: Ehh, och gjort ett eget konsultföretag innan då också, och sen. Just den här branschen, bank och finans, är egentligen helt nytt, helt nytt för tre månader sen kan man säga.
125	Jonas, Jakob: Mm
126	Bank: Ehh, man ser ju också att, att det inte är så himla. Ehh det är klart att det är några bitar som är speciellt här, jag menar några av produkterna och tjänsterna som vi erbjuder är speciella för branschen.
127	Jonas: Mm
128	Bank: Och även det kravet på säkerhetstänkande och integritetstänkande kunder och sånt där. Är väldigt viktigt här.
129	Jonas: Mm
130	Bank: Som inte är alls på samma nivåer hos ett industriföretag. Där är det ju annat som är viktigt.
131	Jonas: Mm
132	Bank: Ehh. Men jag tycker allra mest så är det likheter ju mellan andra tjänsteföretag, större industriföretag då med mycket tjänstemän.
133	Jonas: Mm
134	Bank: Jag tycker mig tanken kring IT verksamhet är väldigt olika. Man ser t.ex. en sådan faktor som ett industriföretag då som pumpar ut produkter hela tiden, som då om systemen ligger nere några minuter så kan man direkt räkna det i tusenlappar vad det kostar.
135	Jonas: Mm
136	Bank: Att hela produktionen stannar av och det är ju samma sak här ju. [Här väljer vi att inte transkribera en mening då det anses vara känslig information, inte heller berör det uppsatsens ämne]
137	Jakob: Mm
138	Bank: Vilka problem det ställer till hos kunderna och alla interna system har ju funkat perfekt under tiden. Men om man säger att det interna systemen ligger nere under tio minuter då är det ju stopp där ute, ingen kund kan expedieras, eller ja så.
139	Jonas: Mm
140	Bank: Det är ganska viktigt just de här bitarna och ju större verksamheten blir desto viktigare får det här eller att det bara funkar hela tiden de här systemen.

141	Jakob: Mm
142	Bank: Det finns inga utrymmen för fel där.
143	Jonas: Mm ok.
144	Jakob: Det är faktiskt någonting vi, vi tänkte kolla lite på i den här uppsatsen då nu när vi är med en bank och pratar, så ska vi kolla även lite andra branscher så att säga. Och se om vi kan se några skillnader eller likheter.
145	Bank: Mm
146	Jakob: Som du säger, nu fick vi kanske en liten hint om det faktiskt finns en del likheter mellan olika branscher.
147	Bank: Jaa, jag tycker nog det att det är mest likheter.
148	Jakob: Mm. Vi får väl se vad vi kommer fram till.
149	Bank: Ja. Haha
150	Jonas, Jakob: Haha
151	Jakob: Ja, om vi skulle ta och börja här med en fråga då.
152	Bank: Ja
153	Jakob: Om vi försöker hålla begreppet öppen programvara i huvudet för det är de flesta frågor handlar om så vi slipper säga det hela tiden.
154	Bank: Ja
155	Jakob: Ehh, jag tänkte fråga. Har du någon uppfattning om dom, den funktionalitet, alltså dom funktioner som erbjuds av öppen programvara motsvarar organisationens behov här.
156	Jonas: Jag ska bara klämma in här också. Är det, på nåt, det är ungefär 20 frågor vi kommer riva av nu.
157	Bank: Ja
158	Jonas: Är det på någon fråga du inte har någon uppfattning så är det liksom helt ok.
159	Bank: Ja
160	Jonas: Det är en del av, av vårt resultat också att många saker har man inte en uppfattning om.
161	Bank: Ja, precis. Amen som sagt, när jag tänker öppen programvara, alltså, mina, de jag kan relatera till, det är åter då klientprogram som Office paketet eller databas program
162	Jonas, Jakob: Mm
163	Bank: Så det är det jag relaterar till, eller hur man ska säga
164	Jonas: Mm
165	Bank: Hur var frågan?
166	Jakob: I fall funktionerna i de här programvarorna motsvarar behoven i organisationen så att säga.
167	Bank: Ja funktionsmässigt så är min uppfattning att det gör det säkert, det för de
168	Jakob: Mm
169	Bank: Om man bara skulle titta funktionsmässigt
170	Jonas, Jakob: Mm
171	Jakob: Mm
172	Bank: Om man skulle göra kravspecifikationer som jämföra funktionskravslista så tror jag nog att

173	Jakob: Mm
174	Bank: tror jag säkert
175	Jonas: Ehh, man kan ju säga att det finns en skillnad mellan att funktionen finns och att den utför samma resultat.
176	Bank: Mm
177	Jonas: Tror du att, hur, vad är din uppfattning om resultatet som den öppna programvaran genererar?
178	Bank: Ja, ja, har jag en uppfattning där? Alltså vad min känsla är att resultatet är? Lika bra eller lite sämre, är min känsla.
179	Jakob: Mm, ehh, ehh, har du, det blir mycket uppfattningar här, men ehh, har du någon uppfattning om låt säga att ni skulle välja ett alternativ, öppen programvara.
180	Bank: Mm
181	Jakob: Ehh, de, öppenprogramvaras förmåga att samverka med dom andra existerande programvarorna som ni har idag?
182	Bank: Ja
183	Jakob: Tror du att ni skulle, skulle det vara möjligt eller skulle det bli någon konflikt eller finns det några sådana tankar.
184	Bank: Ja, alltså, det är ju alltid ett problem. Eller problem och problem men något som skapar väldigt mycket arbete att integrera våra program
185	Jonas: Mm
186	Bank: Och det tror jag alla organisationer har problem med, eller problem och problem, men det är en stor utmaning, det ligger så stor vinning i att koppla ihop saker och ting.
187	Jonas, Jakob: Mm
188	Bank: Att få hela processen att flyta på, det är ännu långt in idag, och det trodde jag för många år sedan att man skulle kunna komma förbi på något slags sätt, men så är det. Medarbetarna har flera inloggningar till olika system och man gör delar av sin process och till viss del överlappar det också, man måste göra om samma sak igen i nästa system.
189	Jakob: Mm
190	Bank: Det är klart att de är en jättestor grej där att kunna knyta ihop processerna
191	Jonas, Jakob: Mm
192	Bank: Och då även att integrera programmen så att det fungerar
193	Jakob: Mm
194	Bank: Och då, har vi ju mycket och mycket förlitat oss på Microsofts teknologi, för att faktiskt kunna knyta ihop saker.
195	Jonas, Jakob: Mm
196	Bank: Ehh, till exempel i portaler, så med Microsoft sharepoint
197	Jonas: Ja, just det
198	Bank: Och så finns det olika integrationshjälpverktyg där också för att få det att flyta ihop, så det är klart, när vi då har valt, i vissa fall andra Microsoft program så ser man ju då hur väl det fungerar.
199	Jakob: Mm
200	Bank: Man får ju även mer sådant som man frågade om, i det hela, medan i andra fall då med mer udda program som ska integreras med varandra så blir det större jobb, såklart.

201	Jonas, Jakob: Mm
202	Bank: Och det, större jobb, så fort det är någonting så, det drar väldigt snabbt iväg och blir mycket pengar utav det. Så väldigt snabbt så kostar det många gånger mer än licenskostnaden då.
203	Jakob: Mm
204	Bank: En, vad ska man säga, en hörnpelare här i våran strategi är ju att göra så lite egen utveckling som möjligt och köpa så mycket färdigt som möjligt.
205	Jonas, Jakob: Mm
206	Bank: Och ska man kunna köpa så mycket färdigt som möjligt, så måste man kunna koppla ihop det med det andra, så långt som möjligt också. Annars för vi ju även kostnader där ju för att få ihop det här.
207	Jonas: Mm
208	Bank: I slutändan.
209	Jonas: Ja okej, ehh, har du någon uppfattning om hur säkerheten är i öppen programvara, så att obehöriga inte, ehh, tillgång till data som de inte borde få, jämfört med då något annat val eller alternativ?
210	Bank: Ja, Mm, alltså för där, säkerhet kan man se på många olika sätt också, om man säger, om man läser i tidningarna, var det finns säkerhetshål och så, så tänker man ju på Microsoft programvara.
211	Jonas: Mm
212	Bank: Men det är antagligen, som jag ser det, därför att det sker mest attacker där.
213	Jonas: Ja
214	Bank: Och, ehh, för våran del, så kan vi bara förlita oss på det som är väl beprövat. Vi tänker inte gå först i något håll i vad det gäller något område, inte säkerhet heller. Utan vi går på de områden där vi vet att andra har testat och det har fungerat väl. Så vårat huvudbanksystem så har vi ju valt ett samarbete med ett stort danskt företag. Ehh Där 160 andra banker har samma system.
215	Jonas: Mm
216	Bank: Som har gått före och tagit smällarna där, i långt och... ehh såhär långt så har det visat sig fungera väldigt väl, men... Så jag tror att, min uppfattning, eller min känsla är också att, att annan mer beprövat teknik såklart är säkrare, men att det kanske sker mer attacker mot de här mer säkrare områdena.
217	Jonas: Mm
218	Bank: Om det nu är så i själva verket det vet jag ju inte
219	Jakob: Nä men det är väl som sagt, de är inte intressant, eller liksom
220	Bank: Nä nä nä
221	Jakob: Det de faktiskt är, är din uppfattning. Ehh, ifall vi skulle gå över lite till vad vi kallar tillförlitlighet
222	Bank: Mm
223	Jakob: eller reliability på engelska, ehh, har du någon uppfattning om antalet fel, alltså oväntade fel som orsakar avbrott i programvara som öppenprogramvara då, ehh frekvensen avbrott på grund av fel, i fall det skulle va, eller om där är mycket fel som dyker upp sådär eller om det flyter på som det ska så att säga.
224	Bank: Ja, jag har, näe, jag har väl egentligen inte någon uppfattning. Jag har bara läst då att det finns studier som visar att vissa olika program med öppen källkod då skulle vara mer tillförlitlig där.
225	Jakob: Mm
226	Bank: Men min erfarenhet säger ju då ändå att, ehh, och de kunderna som jag haft när jag drivit konsult verksamhet och likadant som jag känner för den här verksamheten är att det finns andra aspekter i det där också, den dag då det inte fungerar. Då är det jätte viktigt att det finns, att vi kan lita på att det finns folk som kan den här tekniken som vi har valt.

227	Jonas, Jakob: Mm
228	Bank: Och står man då där och har valt ett smalare spår, så, ehh. Ja då står man ju där också och är väldigt beroende av några få personer som, ehh, som ska lösa det här i slutändan.
229	Jakob: Mm
230	Bank: Så, tillförlitlighet är egentligen mer än bara driftavbrottet, det handlar ju om att, att det är en, en så mogen teknik så att det runt omkring är uppjobbat med rutiner för vad man ska göra i olika scenario.
231	Jonas: Mm
232	Bank: Och förebyggande arbete för, för de här driftavbrotten.
233	Jakob: Mm
234	Bank: Alltså, jag tror att egentligen är det, jag är inte riktigt fördjupad i vår statistik vid driftavbrott, men jag tror nog att det är så att, att det inte är själva, ett system i sig som drabbas, utan det är hela lapptäcket med allting tillsammans.
235	Jonas: Mm
236	Bank: Och då handlar det om hela organisationen runt omkring hur man arbetar med det här.
237	Jakob: Mm
238	Bank: Ehh, hur, ja, hur organisationen är uppbyggd, hur man ska hantera ett fel och hur man går till väga och hur väl man känner till olika feltyper och åtgärder och vad förebyggande arbeten har sagt
239	Jakob: Mm
240	Bank: Så jag tror egentligen inte att det, ja, alltså jag känner mig säkrare i en värld med mer beprövade produkter.
241	Jonas: Mm
242	Jakob: Mm
243	Bank: För då kan man bygga upp allt det här runt omkring, än med öppen källkod där det är mindre beprövat.
244	Jakob: Mm
245	Jonas: Om man tittar på en situation där, om det uppstår ett fel, har du någon uppfattning om, låt säga att det kommer upp någon felruta eller någon liten krasch i programvaran. Har du någon uppfattning då om hur öppen programvaras förmåga är att kunna fortsätta trots ett fel, alltså kunna behandla ett fel väl?
246	Bank: Nä det har jag väl egentligen ingen uppfattning om.
247	Jakob: Jag tror faktiskt du lyckades svara på en annan fråga här också
248	Bank: Ja
249	Jakob: i bara farten, ehh, just det här med att, i fall det skulle krascha, då en programvara.
250	Bank: Mm
251	Jakob: Ehh, den, de antalet resurser som måste sättas in form av tid, personal och så vidare för att återställa till stadiet innan kraschen då så att säga.
252	Bank: Mm
253	Jakob: Och om jag förstod det rätt så, så ju fler som använder det, desto mer rutiner finns det för att kunna hantera problem, kanske i form av support?
254	Bank: Ja precis, och den dagen våra egna resurser inte räcker till så kan man ju tänka sig att det finns andra där ute.
255	Jonas, Jakob: Mm

256	Bank: Som, mycket av våra kritiska system är kopplade till Microsoft på ett eller annat sätt.
257	Jakob: Mm
258	Bank: det är klart, dom har ett helt nätverk runt hela jorden med experter som kan kopplas in där.
259	Jonas, Jakob: Mm
260	Bank: och egna utbildningar och licensieringar och så vidare, så jag menar många av dom som jobbar hos oss, hos [företagsnamn] och utför det här har ju, är ju, licensierade eller vad det nu heter för olika varianter.
261	Jonas, Jakob: Mm
262	Bank: och det är faktiskt viss trygghet att man vet att de kan en viss mängd där
263	Jonas, Jakob: Mm
264	Bank: och i det fallet det inte räcker till så finns det många andra konsultföretag där ute att lyfta in resurser ifrån, och Microsoft själva som besitter den kompetensen.
265	Jonas: Mm, precis
266	Bank: Och den dagen det inte fungerar, [Här väljer vi att inte transkribera en mening då det anses vara känslig information, inte heller berör det uppsatsens ämne]
267	Jonas: Mm
268	Bank: Så har vi varit väldigt glada, då i allt det här, så har vi varit glada att det funnit människor som kunnat hjälpa oss med det.
269	Jonas: Mm
270	Bank: Så det är ju verkligen en förtroende fråga i slutändan att, att, veta att det här problemet vi har, de har Microsoft sett i 38 fall i något annat land, någon annan stans, och har faktiskt en plan. Att vi kan göra på det här sättet för att avhjälpa det
271	Jonas: Mm
272	Bank: Det kan bli mål att det tar 3 dagar innan vi är uppe igen, och vi kan liksom kommunicera ut, och på platsen står då
273	Jonas, Jakob: Haha
274	Bank: Så kan man stärka lite, vi vet inte riktigt, det är väl på några veckors tid kanske då, om vi har tur.
275	Jonas, Jakob: Haha, Mm
276	Bank: Det är så mycket runt om kring som hänger samman som gör, som är viktigt inom det hela
277	Jakob: Mm det kan ja tänka mig
278	Jonas: Om vi tar och tittar lite på användbarheten
279	Bank: Ja
280	Jonas: Ehh, så om man tittar på strukturen i ett program, hur det är uppbyggt med, ja t.ex. menyer och såhär.
281	Bank: Mm
282	Jonas: Har du någon uppfattning där, hur det är, om det är bättre eller sämre i öppenprogramvara.
283	Jakob: Det handlar väl mest om det är lätt eller svårt att förstå hur det är.
284	Bank: Ja
285	Jonas: Mm, ja just det

286	Bank: Ja, just det
287	Bank: Att det är logiskt och intuitivt och så?
288	Jakob: Mm
289	Bank: Njea, ja, det är kanske egentligen. Egentligen kanske det kan vara så att det är på samma nivå. Men sen, nästa lager på det handlar ju om hur väl utbildad man är i sammanhanget ju.
290	Jonas: Mm
291	Bank: Att du vet man direkt att, den här menyn går när jag vill göra det här
292	Jonas: Mm
293	Bank: Alltså, det som känns självklart kan ju vara att, det har man lärt sig längs vägen så blir det ju självklart också
294	Jonas: Mm
295	Bank: Ehh, och sen, ja, nu känns det som om jag pratar om Microsoft hela tiden här då men, men det är klart att, jag tänker på att ju, ju fler. Ju större grupp användare man får ut sin programvara till, desto mer input kan man ju få från olika, olika typer av användare.
296	Jonas: Mm
297	Bank: Så kan man skapa sig ett program som är mer eller mindre optimalt för att använda i slutändan. Och det kan man kanske göra om man når en inre grupp med användare också.
298	Jonas: Mm
299	Bank: Men min uppfattning är nog ändå att de här mer större kommersiella produkterna dom, dom är lättare att använda.
300	Jonas, Jakob: Mm
301	Jonas: Hur tror du det är att lära sig själva produkten, tror du det är lättare eller svårare att lära sig öppen programvara.
302	Bank: Om man står på noll i båda fallen tror jag det skulle vara lika lätt eller svårt:
303	Jakob: Mm
304	Jonas: Okej
305	Bank: Men det är ju att man har ju så mycket annat med sig, troligtvis, som gör att man inte står på noll
306	Jonas: Ja just det
307	Jakob: Ehh, har du någon uppfattning om, om, ehh, alltså attraktivitet, det är väl snarare rent estetiskt alltså, ehh, upplevelsen man får av att använda till exempel en Microsoft produkt eller öppen programvara, alltså dess förmåga att tilltala användaren.
308	Bank: Mm, ja, jag vet egentligen inte, kanske ungefär lika bra
309	Jakob: Mm
310	Jonas: Ja, den andra vi väl fått in ändå, tycker jag
311	Jakob: Ehh, en annan grej här då som kanske handlar lite mer, ehh, inte så mycket det som kanske, ehh lite mer interna processer eller vad man ska säga i programvaran, som handlar om svars och bearbetningstider för en exekvering av, alltså, en, om du gör en uppgift, hur lång tid det tar för att få den utförd och så vidare.
312	Bank: Ja
313	Jakob: Ehh, upplever du där att öppen programvara skulle, motsvara era behov så att säga.
314	Bank: för svarstider?

315	Jakob: Ja
316	Bank: Mm, jaa, det skulle det säkert, alltså den enskilda installerade delen så då, skulle det säkert, är min uppfattning.
317	Jakob: Mm, ehh i fall vi går in på och tittar på resurser som programvaran kräver, liksom hårdvara, material och annan, kanske mjukvara som man måste kombinera, vilken personal som måste vara på plats för att det ska fungera och så vidare.
318	Bank: Mm
319	Jakob: Finns det någon uppfattning där kring öppenprogramvara, jämfört med, får det väl nästan bli
320	Jonas: Mm
321	Bank: Ja, men jag förstår, ja. Då tänker jag såhär, då borde det röra sig om t.ex. en databas applikation till exempel, en databas.
322	Jonas: Mm
323	Bank: Ehh, att jag tror att, ehh, om man då skulle jämföra de här programmen med varandra, då är min uppfattning att, dom tekniska kraven på systemen, då skulle säkert de här som är öppenprogramvara, skulle säkert, klara, klara bättre än andra som kommersiella produkter, med svarstid, och ehh, driftstabilitet, om det är ett begrepp.
324	Jonas: Mm
325	Bank: Haha, ehh och andra sådana prestandabegrepp, så tror jag att det skulle stå sig väl där.
326	Jonas: Mm
327	Bank: Men, igen då då, så tror jag att det finns andra områden som skulle göra tillförlitligheten ännu större hos den kommersiella produkten, totalt sätt, som till exempel, om man säger vår kärnapplikation, är såklart uppbyggda kring, en databas eller ett antal databaser som lagrar all information om kunder och konton, och allt vad det är.
328	Jonas, Jakob: Mm
329	Bank: Ehh, som många sådana här administrativa system, såklart är uppbyggt kring databaser. Ehh och det är klart att, hela vår verksamhet, vilar ju också på det där. Så valet då att välja en mindre, något mindre beprövat mot något mer beprövat, alltså valet är inte speciellt svårt då. Även om det, även... Även där då medför större licenskostnader, och det säkerligen är så att det kräver mer kapacitet i from av hårdvara också.
330	Jonas: Mm
331	Bank: Produkterna alltså... Så våran upplevelse av stabilitet och tillförlitlighet är ändå högre hos de här kommersiella produkterna
332	Jonas: Mm
333	Bank: kan vi säga, men det är inte säkert att det behöver vara så i praktiken men, men. Klart att det hör ihop med, jag nämnde ju det här, det som finns runt omkring också i vår organisation, att hur väl då, vi som ska besluta om det här uppfattar att vi har resurser som klarar av det ena till det andra.
334	Jakob: Mm
335	Bank: Att... Det är just att, det är mycket kring det att konfigurera det så att det passar oss såväl som möjligt, ehh, och hantera administrationen som finns runt omkring på enklast möjliga sätt som, användare och behörigheter och allt det kan handla om som är kopplat ihop med det där.
336	Jakob: Mm
337	Bank: Att... Så det är klart att det i sig blir en rätt så stor kostnad också för en sådan här organisation, det är ett par stycken som bara jobbar med att bara sätta upp behörigheter.
338	Jonas: Mm
339	Bank: Det är klart, skulle det behövas ett par stycken till så kan man ju räkna ut direkt vad det kostar.

340	Jakob: Mm
341	Bank: ...om det förenklas
342	Jonas: Mm
343	Bank: Men då kan man ju jämföra då med ett, ett, en mindre organisation, då är det inte en så viktig aspekt längre, så då, då tar det bara en del av någons persons tid och man kan ändå ha grepp på helheten genom att bara ha det i huvudet. Man kommer förbi något stadium där det liksom några steg uppåt så, som det måste systematiseras, allt det där va
344	Jonas: Mm
345	Bank: Man jobbar uppikring administrationen av det hela
346	Jonas, Jakob: Mm
347	Bank: Och det är då det kring rutinerna som, som, ja egentligen blir det största delen av kostnaden
348	Jonas: Mm, ehh
349	Bank: Nu vet jag inte exakt vad så, vad själva licenskostnaden är men, den kan vara några miljoner för oss kring databasapplikationerna, men behöver man ha 5 personer till, då blir det ju 5 miljoner till, direkt.
350	Jonas, Jakob: Mm
351	Bank: Och så är det... ganska snabbt så blir det att man börjar ha, är det körigt att hantera så behövs det några till att hantera det.
352	Jonas: Mm
353	Bank: Och det är klart, är det ett företag med 10 000 anställda, då har ju det här, blivit ännu mer problem, i skalan, förstås
354	Jakob: Mm
355	Jonas: Precis
356	Jakob: Alltså, så det finns någon form av skalbarhet i kostnaderna. Det är en intressant tanke, jag tänkte bara höra, hur många personer berör det när ni sitter och diskuterar en strategi, där ni ändå har en gemensam linje även om den inte gäller alla mer specifikt så, hur många personer är det som berörs av de här besluten?
357	Bank: Mm
358	Jakob: Jag har för mig du sa en siffra innan men den försvann ut i...
359	Bank: Ja, det beror på vad det är och så men... [företagsnamn], vi är väl en 330 anställda
360	Jakob: Mm
361	Bank: Och på [företagsnamn], vad kan det vara? 250 åtminstone.
362	Jonas: Mm
363	Bank: och på [företagsnamn] är det 120 anställda, och alla användare har ju Officepaketet.
364	Jonas: Mm
365	Bank: Skulle vi nu ändra det så skulle det vara en gigantisk operation såklart
366	Jonas: Mm
367	Bank: eller byta mailsystem till exempel, eller vad det nu kan vara för någonting
368	Jonas: Mm, ja precis, vi kan ta...
369	Bank: Medan annat när det gäller databas applikationer så, om man nu gör det så att inte våra slutanvändare här

	störs av det så, så, klart rör det, det här IT-teamet som ska administrera och förvalta det där i slutändan.
370	Jonas, Jakob: Mm
371	Bank: som det är, då kan det va kanske vara mer 25 personer det berör då
372	Jonas: Det är också en väldigt intressant aspekt där, tror du att, ehh, hur omfattande ett sådant byte skulle vara till öppen programvara, från det ni har idag, skulle det vara väldigt resurskrävande eller...
373	Bank: Det skulle vara väldigt resurskrävande.
374	Jonas: Ja
375	Bank: Vad det än är, alltså, om det inte är, även om det är något av våra kärnsystem så är det klart att, det kan man ju säga att allting som våra slutanvändare använder, eller de här huvudsystemen och såhär. Då skulle det vara väldigt, väldigt, resurskrävande. Men det är klart, skulle det vara en liten isolerad applikation, en del av det hela så skulle det vara mindre resurskrävande
376	Jonas: Mm, precis
377	Bank: Om det är något enstaka som berör 10 personer så är det klart att då är det inte alls något problem, men så är det liksom.
378	Jonas: Näe, om vi tittar lite på förmågan att underhålla en programvara, ser du, ehh, att det är lättare eller svårare på något sätt att hitta, om det skulle vara ett fel, och identifiera felet i öppen programvara.
379	Bank: Jaa, det är klart... För vår del tror jag inte att det skulle vara det egentligen, för att vi har strategin att vi ska egentligen inte pilla med det på så djup nivå.
380	Jonas: Mm
381	Bank: Men jag kan tänka mig att andra verksamheter, att det kan vara viktigt för andra typer av verksamheter. Att vara mer, om man har det behovet att vara och pilla i det, om man säger att man, kompletterar systemet med något egenutvecklat system till exempel
382	Jonas, Jakob: Mm
383	Bank: Då har man nog behovet av att mer kunna felsöka direkt i själva koden i det där...
384	Jonas, Jakob: Mm
385	Bank: datorprogrammet... då också, men, ehh. Ja ja, vi i och för sig har ju också så, i vissa fall så är det sådana kritiska delar, och det är sådant som inte finns på den öppna marknaden att köpa in så där tvingas vi då att köpa egen utveckling av det hela.
386	Jonas: Mm
387	Bank: Så där kanske det skulle kunna vara en faktor, just i de enskilda fallen, eventuellt då.
388	Jonas: Mm
389	Bank: För det andra då så, de ska va standardprogram så långt som möjligt, där, där är det inte för oss viktigt, det kan ju vara viktigt kanske för vår leverantör, om de bygger sitt program delvis på någonting som har öppen programkod.
390	Jonas, Jakob: Mm
391	Jakob: Så kan man då säga att, ehh..., jag vet inte om jag lägger orden i munnen på dig här. Men du anser att det är lättare att modifiera öppen programvara.
392	Bank: Ja det tror jag att det är
393	Jakob: Är det så?
394	Bank: På, på ett sätt medans, ehh, för vår del då, i vår verksamhet.
395	Jakob: Mm

396	Bank: Då modifierar vi ju det mer på ett, ehh, på en högre nivå på det att. Ehh jag menar vi ändrar vad ska man säga, ja men för slutanvändaren, då är det sådana där små enkla grejer som att ändra vilken ordning kolumnen ska vara i Outlook till exempel. Eller vilka, vilka fält som skall finnas med i kontaktkortet i Outlook och på den nivån.
397	Jonas, Jakob: Mm
398	Bank: Och då är det klart att, då är ju ett sådant program som Outlook, ganska lätt att arbeta med. För där kan man faktiskt hitta, om man är lite mer än normalanvändare, så kan man faktiskt modifiera det.
399	Jonas: Mm
400	Bank: någorlunda... Ehh, eller bestämma hur man ska exportera information och så. Och vi har ju, men det är kanske på den nivån vi har behovet då att modifiera.
401	Jonas: Mm
402	Bank: Men alltså om man bara skulle bygga, skulle utveckla ett eget delprogram och koppla ihop med något annat program som vi köper in, så är det klart då skulle vi ha behovet att modifiera på programkods nivå och då skulle det säkert vara enklare.
403	Jonas: Mm
404	Bank: Men...
405	Jonas: Om vi säger att ni tar in en, programvara, eller uppgradering, eller en modifierad programvara och i så fall antagligen vill testa den
406	Bank: Mm
407	Jonas: Hur ser du då möjligheterna att just kunna göra det jämfört med något annat alternativ
408	Bank: Med att, att, åtkomsten till att testa programmen innan vi köper in dom och så.
409	Jonas: Mm, ja precis
410	Bank: Ja, Mm, Ah, det är nog ungefär, sammantaget tror jag att det skulle vara det samma, för. Alltså för, när vi ska testa en ny produkt, som vi gör egentligen hela tiden för att utvärdera olika saker och ting.
411	Jonas: Mm
412	Bank: Så handlar det om för oss ju som är, skall vara användaren där, oftast, problemet där är att, Vi har en bild i huvudet, ja men så... den här uppgiften vill vi utföra. Men det, det blir ju ett rätt stort problem för oss då som är användare av det hela och sen gå in i en applikation och sen, kunna testa den och kunna se om vi kan utföra våra uppgifter med applikationen. Alltså, det är oftast ett jätte stort steg och där kräver man, eller det krävs att det skall bli någonting av det. En leverantör av programvaran som kan ta våra abstrakta krav och tänka om att, om ni vill utföra det här...
413	Jonas: Mm
414	Bank: Det skulle man kunna göra på det här sättet med de här 10 stegen i programmet.
415	Jakob: Mm
416	Jonas: Okej, ehh
417	Bank: Vi behöver mer en speaking partner för att...
418	Jonas: okej
419	Bank: kunna förstå det, eller, och det inte säkert att vi förstår det ändå, ifall, och kan jämföra två applikationer. Vi har gjort massor av gånger och köpt in system och så, det är jätte svårt att sitta och försöka få in om programmet kan uppfylla det man vill.
420	Jonas, Jakob: Mm
421	Bank: Ehh, Mm, det är klart vi har förväntningar också på ett program som man kanske tror bara skall följa med också, så det är...

422	Jonas: Mm
423	Bank: Grunden i det är att vi måste ha trovärdighet för leverantörerna, att det är ett sånt spritt program och att det används i liknande branscher, så då kan vi förmoda att det har en utskriftsfunktion.
424	Jakob: Mm
425	Bank: Och att det har... och det kanske inte räcker ens där, vi... Innan vi tar beslutet, då vill vi åka till en annan kund som har det.
426	Jonas, Jakob: Mm
427	Bank: Dom har ju exakt samma problem som oss, med den kundens ord kan säga att vi kan utföra det här. Att vi, liksom mer jämför programvaran på det sättet än att gå in och titta på de enskilda funktionerna, för det är för komplicerat för oss.
428	Jonas, Jakob: Jaa
429	Bank: Sen när vi ska köra in det, då är det ett långt implementationsprojekt och det innefattar mycket människor och det ska utbildas och allt. Och egentligen slutet av det kan vi säga om vi gjorde rätt val ju.
430	Jonas, Jakob: Mm
431	Bank: Vi måste lita på faktorer, på mer, på högre abstraktionsnivå i början, innan...
432	Jonas, Jakob: Mm
433	Bank: och sånt
434	Jonas: Okej
435	Bank: Och det handlar, säkert ofta om hur duktiga leverantörerna är på att transformera då, kundens abstrakta krav till någonting... funktionella, funktionell påverkan i systemet, tror jag.
436	Jonas: Mm
437	Bank: som påverkar valet. Och så tror jag i mer begränsade smalare system så skulle det kanske kunna vara så att någon tekniker hos oss, tar in några olika system och själv testat ifall... Men det är nog i undantagsfallet skulle jag vilja säga, för våran del.
438	Jonas: Tror du att en sådan uppdatering som kommer till en öppenprogramvara, att den på något sätt kan ha önskade effekter?
439	Bank: Mm, så skulle jag tänka i alla fall, att den kan ha det, men det vet man ju inte. Men då är det skönt att höra någon annan kund säga det.
440	Jonas: Mm
441	Bank: Att ja när vi körde in det här, då hände också det här.
442	Jakob: Mm
443	Bank: Så kan man förbereda sig lite på det
444	Jakob: Mm
445	Jakob: Ehh
446	Jonas: Då har vi nog kört alla frågor där va?
447	Jakob: Mm, ja
448	Jonas: Ehh, det finns en annan faktor inom mjukvarukvalitet som kallas flyttbarhet, och ehh
449	Bank: Ja, vad innebär det då?
450	Jonas: Det innebär lite att man skall kunna flytta programvaran från en miljö till en annan till exempel, fast det är bara en aspekt.

451	Bank: Mm
452	Jonas: av det... Och en annan är då att man skall kunna anpassa, då snabbt, efter er verksamhet.
453	Bank: Mm
454	Jonas: Tror du att öppen programvara, att den är så anpassningsbar för er verksamhet.
455	Bank: Mm, kan den säkert vara i vissa fall, kan den nog vara.
456	Jonas: Om det liksom skulle vara lätt
457	Bank: Mm, det är om man tänker ett fall att det inte går att köpa en helt färdig produkt för det som vi vill utföra utan...
458	Jonas: Mm
459	Bank: Utan vi vill ha det specialdesignat och då vi vill vi ha det så snabbt som möjligt, då kan man ju tänka sig att kostnaden blir så låg som möjligt, projekt tiden blir kortare och så. Jag vet inte om det skulle spela någon roll om det är öppen programvara eller inte, men då skulle man mer få en leverantör som kommer in och säger dom såhär. Ja vi har det här systemet och det är som en verktygslåda och vi skruvar till det lite åt er och så.
460	Jonas: Mm
461	Bank: På något sätt och så... Men ja, jag vet inte, om det, men det kanske man skulle kunna göra med andra produkter också som inte är öppen källkod och så...
462	Jonas: Ja
463	Bank: Och vissa, vissa produkter som inte är opensource, ger ju också våran leverantör tillgång till källkoden.
464	Jonas: Mm
465	Bank: Om våra leverantörer har rätt licensiering, så får dom då böka omkring i det där så gott dom kan då.
466	Jonas, Jakob: Mm
467	Bank: Så man kan ju uppnå det ändå, men, man, ja, jag vet inte om det ska ha någon... Om de skulle skilja sig egentligen från min uppfattning.
468	Jakob: Mm, Ehh, en annan punkt som ligger under flyttbarheten handlar om själva installations, ehh, förfarandet att man... själva bara att installera öppen programvara, ehh. Då ska man nog tänka bort alla konsekvenser det får med att ersätta osv., utan bara själva installationen av programmet så att säga.
469	Bank: Ja
470	Jakob: Har du någon uppfattning i fall det skulle vara ett krävande arbete, jämfört med då får det nog bli då skulle jag nog tro...
471	Bank: Mm, andra produkter...
472	Jakob: Ja precis.
473	Bank: Jaa, Mm... Jag vet inte riktigt. För våran del så har vi ju, ja vad kallas det, ett distribuerat arbetssätt kanske det kallas, ja jag vet inte vad det kallas. Men vi gör själva installationen centralt och sen trycker vi ut det på alla klienterna.
474	Jonas: Mm okej
475	Bank: för att det skall vara effektivt
476	Jonas, Jakob: Mm
477	Bank: Det är klart att, i det fallet det inte går att göra på det sättet då måste ju någon person fysiskt åka runt till alla ställen och installera programmet.
478	Jonas: Mm

479	Bank: Sen kommer en ny version, och då måste man åka ut igen.
480	Jakob: Mm
481	Bank: och är det fel på det, då måste man åka ut då också, och så. Så det är klart, det är en viktig aspekt då, att man kan göra det med rätt program.
482	Jonas, Jakob: Mm
483	Bank: Ehh, och, ja, att... Det är egentligen lika dant där ju... Om man tänker då på den enskilda teknikern som skulle göra det ena respektive det andra så kan man ju tänka sig att... Ja men vi vet ju det där med installation då ju... Det ska ju gå enkelt men så dyker det alltid upp olika problem längs vägen.
484	Jakob: Mm
485	Bank: Hehe, Ju fler gånger man gjort det i desto fler olika miljöer, speciellt sådana miljöer, speciellt sådana som liknar våra. Då kan man kanske anta då att det blir färre problem, och det går lättare.
486	Jonas: Mm
487	Bank: och leverantörerna har tänkt in olika scenarier som då är, som de klarar av. Men jag uppfattar det då mycket som om man väljer det här programmet som har öppen källkod, att mycket av ansvaret hamnar liksom på oss. Att ta hand om de här och lösa de här problemen. Och, det vill vi helst inte ta, vi vill ju helst ha en leverantör som tar det ansvaret.
488	Jonas, Jakob: Mm
489	Bank: Som vi kan skjuta tillbaka problemen till. Men jag vet inte om det är så, men jag upplever att det är så, speciellt för vår del då.
490	Jonas: Mm
491	Bank: Och att det på det sättet skulle kunna bli krångligare
492	Jonas: Mm
493	Bank: Så att... På det sättet.
494	Jonas: Då är vi nog klara där, för den här har vi redan behandlat tycker jag.
495	Jakob: Ja, precis, ja det var nog de punkterna vi hade så att säga.
496	Bank: Ja
497	Jakob: Så ja. Vi är väldigt tacksamma att du ställde upp på det här
498	Jonas: Väldigt tacksamma!
499	Bank: Hoppas att det är till någon hjälp för er.
500	Jonas: Absolut
501	Jakob: Det tror jag, spontant så känns det så.

8.4.2 Produktutvecklingsföretag

1	Jakob: Ja, om vi ska börja lite kort med att förklara vad det är vi tänkt göra här, så du får en förståelse för...
2	M: Vilken linje går ni på?
3	Jakob: Systemvetenskapliga programmet
4	M: Ja...
5	Jakob: båda två... eh... på kandidatterminen...
6	M: Ja...
7	Jakob: Så det e väl det sista nu innan kandidatexamen, ifall vi nu får det här avklarat
8	M: Ja...
9	Jakob: det får vi se...
10	Jakob: Vad har du själv för utbildning?
11	M: Jag är matematiker
12	Jakob/Jonas: Mm
13	M: ... i botten, men jag har ingen såndär linje utan jag har enstaka kurser i botten, men jag har en fil.kand. i matematik.
14	Jakob/Jonas: Mm
15	M: Så det e lite ovanligt kanske men...
16	Jakob: Det fungerar?
17	M: Ja
18	Jakob: Mm, ehh uppsatsen som vi skriver handlar ju om öppen programvara då, eh..., och den handlar väl egentligen om att försöka fånga en bild av uppfattningar om mjukvarukvalitet i öppen programvara.
19	M: Mm
20	Jakob: Och betydelsen av mjukvarukvalitet när det gäller beslutsfattande kring, eh alltså, vilken programvara man skall använda.
21	M: Mm
22	Jakob: Betydelsen av mjukvarukvalitet vid beslutsfattandet
23	M: Ja...
24	Jakob: Eh, och därför har vi då tagit kontakt med dig för att vi att vi har förstått att du har en insikt i eller har i alla fall varit med i den här beslutsfattandeprocessen vid programvara, när ni har bestämt er för vad ni ska använda
25	M: Ja, alltså det stämmer både egentligen, eller både... delvis... jag har... insikt har jag i vart fall... och jag har varit med i rätt många beslut... om man ska kalla dom beslut... det är inte alltid det är så tydligt uträknat, men processer i alla fall.
26	Jakob/Jonas: mm?
27	M: Och eh... det har jag egentligen både från R & D, alltså produktutvecklingen här och från IS-sidan.
28	Jakob: Mm

29	M: Men nu är det IS som intresserar er vad jag förstår, alltså interna system?
30	Jakob: precis, precis!
31	M: Men vi kör ju... vilket kan vara intressant då som bakgrund för det ger ju ändå en sorts profil för företaget, om vi ska säga det först att vi kör ju... generellt kör vi mycket både open source och egenutvecklat.
32	Jakob: Mm
33	M: I förhållande till inköpt programvara
34	Jonas: Mm...
35	M: Så med den anledningen skulle jag säga, stora och kanske, alltså jag har svårt att säga relativt sett men alltså åtminstone... åtminstone stor... Eh...
36	Jakob: Är det merparten eller e det liksom... det blir kanske svårt att säga?
37	M: Ja det beror på hur man räknar... alltså det är inte så lätt att säga. I pengar räknat så e det ju ingenting alls...
38	M/Jonas/Jakob: *skratt*
39	M: alltså i inköpspris... så att...
40	Jakob: Näe...
41	M: Men eh, det e ju... alltså både vad gäller produkter och vad gäller interna system och verktyg så kör vi mycket både open source och egenutvecklad programvara och vi supporterar också en del open source härifrån.
42	Jakob: Mm?
43	M: Alltså vi maintain... alltså dom som jobbar här e maintainers för många olika programvaror.
44	Jonas: OK
45	M: Det gäller till exempel eh... ett flertal exempel är underhåll utav vår egen kompilator, för vår egen kärna. Den e baserad på GCC som är en öppen kompilatorramverk eller vad man säger... och den genererar... alltså vi använder samma back-end... eller front-end som GCC och sen genereringen av objektкод... eh maskinkod förlåt, förlåt... det är öppen källkod som vi underhåller och andra exempel är ett testramverk för enhetstester för C. Och... ja... det är ett par exempel i alla fall.
46	Jonas: Mm!
47	M: Det är också för att ge lite bakgrund, det är också mycket kan man säga... både på utvecklingssidan, framförallt där, men också på systemsidan alltså på... det är det som intresserar er. Så är det väldigt mycket en eh... kulturellt värde att värdesätta open source och det är egentligen inte förankrat i någon sorts strategi på ledningsnivå kan man säga.
48	Jonas: Nää?
49	M: Alltså, skulle man studera bara det, hur den relationen ser ut mellan de starka värderingar och på strategin egentligen på mellanchefsnivå och på utveckelnivå, kontra på eh... alltså på... när det gäller produkt så är det ju en strategi som är etablerad, men den är egentligen bara baserad på att det funkar... kan man säga.
50	Jonas: Mm
51	M: Alltså, det är värt att säga någonting om det. Fram till slutet av 90-talet så var alla våra produkter baserade på egenutvecklad programvara. Alltså vi hade gjort ALLTING i källkoden i stort sett, med några få undantag. Så det var egentligen en helt annan strategi som då också kanske... alltså vi är inte så ovana vid inbyggda system, men ändå... kanske när det gäller vårt egna operativsystem till exempel, medan de allra flesta kanske framförallt inom [Ohörbart] köper vi kanske in, eller använder öppet...
52	M: Men vi hade allting eget, och det var en ganska stor utmaning för vi... det innebar att vi hade egna implementationer för olika filsystem och så vidare. Det var ganska mycket som vi fick uppfinna, eller inte uppfinna utan återskapa. Eh, sen så e vi då över till att basera produkterna på Linux och det var egentligen bara för att det gick för dom som gjorde det, det var då ganska få personer som fixade med det. Då såg man ju plötsligt att ja då slipper ju eh... då är det väldigt mycket programvara du slipper göra själv.

53	Jonas: Ja just det...
54	M: Så att... det är... och även när det gäller interna system som är lite mer ert fokus så har vi ju... vi har gjort många egna grejer och vi har... alltså senast vi... Häromdan så var jag på ett möte med en beställargrupp från verksamheten som, dom ville ha en viss typ av ett system.
55	Jakob: Mm?
56	M: Och då hade dom med... två av deras krav i den här preli... alltså kravspeccen var för det första bör eller ska vara open source, för det andra, vilket är egentligen lite anmärkningsvärt, ska använda My-SQL som databas.
57	Jonas: Mm
58	M: Vilket ju egentligen ju inte alls bör vara... även om jag e glad för det så... så ska ju inte det va ett krav från en beställare egentligen.
59	Jonas: Nä...
60	M: Men det säger en del om kulturen...
61	Jonas: mm!
62	Jakob: Det är intressant det du säger det här med kulturella, eh kulturella värderingar som handlar även om beslutsfattande kring programvara. Alltså man kanske kan... man brinner nästan för öppen programvara så att säga liksom... att även det kan påverka liksom.
63	Jakob: Men så att säga, eh, e det så att det finns en uttalad policy eller nåt sånt där för vilken typ av programvara som skall användas?
64	M: Nä det gör det ju inte...
65	Jakob: Det är situationsspecifikt?
66	M: Men i praktiken kan man nog säga att vi... Det finns... Man skulle nog kunna säga... Det här borde man ju undersöka för att verkligen... jag är ju bara en informant även om jag har rätt god insyn... alltså... som det här exemplet med beställargruppen visar så är det ju nästan ett default-antagande att det ska va, att vi ska ha... öppen programvara.
67	Jo/Jakob: Mm?
68	M: Eh, både från beställarsidan uppenbarligen men framförallt på [Ohörbart]
69	Jakob: Mm
70	M: Och vad... man kan säga så... Vad har vi för... Vad har vi för? Det e lättare att räkna upp vilka programvaror vi har som alltså inte är fri eller öppen. Det är ju att... förutom en del utvecklare så kör alla Windows XP på desktoppen och dom flesta... Sen är det många som kör dual-boot eller kör flera maskiner, som kör Linux och så en del kör bara typiskt Linux eh ja... i stort sett bara det, kanske nån kör FreeBSD jag vet inte men. Men eh... så man kan säga att väldigt många kör Windows XP på desktoppen.
71	Jonas: Mm
72	M: Och sen har vi en del... och så har vi... då kör dom flesta också Office-sviten fastän dom egentligen skulle behö... alltså egentligen så skulle vi kunna köra en nittio... alltså allting utom Excel skulle ersättas med OpenOffice skulle jag vilja säga, till 99,9%. Även Excel skulle nog kunna ersättas, men just alltså... det är ett för skakigt beslut att ta liksom bara så...
73	Jakob/Jonas: Mm
74	M: Om vi nu inte kör Excel längre, vad kommer då inte funka längre? Det är mycket som hänger på Excel med olika makron och avancerad funktionalitet och så vidare, framförallt på finansavdelningen men också på en del andra sidor.
75	Jakob/Jonas: Mm...
76	M: Så därför blir det rätt mycket de facto-användande av Office.
77	Jonas: Mm

78	M: Och sen... sen är det en del stora och dyra verktyg på egenutvecklingssidan som inte finns...
79	Jonas: Mm
80	M: Det är då dyr ofri programvara kan man säga.
81	Jonas: Mm
82	M: Vårt ERP-system, vi använder IFS, är långt ifrån fritt eller öppet och... vi har ett redovisningssystem som också är inköpt.
83	Jonas: Mm
84	M: Och en del integrationsprogramvaror runt ERP, alltså runt finanssystemet, är också... är också köpe-program.
85	Jonas: Det känns som... känns som du...
86	M: Men det e... och så har vi några Windows... så har vi några Windows-Servers men nästan alla servers är... kör öppen källkod.
87	Jonas: Det känns som du har ganska schysst bild rent allmänt om att ni använder lite öppen källkod, eller öppen programvara för att det är ett komplement.
88	M: Mm...
89	Jonas: Och sen använder ni då till exempel viss windows-programvara för att den inte går att ersätta?
90	M: Det är mer av historiska skäl som gör att vi kör den...
91	Jonas: Ja OK. För att om du lite... vi har ungefär 20 korta frågor...
92	M: Ja, ja! Det va jag som började prata...
93	Jonas: Men det är perfekt för det e verkligen vad vi är här för
94	M: Ja kör dom för det e rätt bra för det är... japp!
95	Jonas: Mm, så om du då tänker dig in allmänt då om öppen programvara, så kommer vi då ställa 20 frågor som handlar mjukvarukvalitet.
96	M: Japp
97	Jonas: Och e det nån du känner att du inte har någon uppfattning om så är det liksom bra, då är det bara att säga det i så fall.
98	M: Mm Mm
99	Jakob: Ifall det skulle vara någon oklarhet i frågorna så får du gärna ifrågasätta frågan så att säga, så att vi kan reda ut det...
100	M: Ja det gör jag gärna!
101	Jakob: Om vi börjar här nu med att om skulle diskutera lite funktionalitet. Om vi skulle se det som ett delbegrepp av mjukvarukvalitet.
102	M: Mm
103	Jakob: Eh, och så... och eh, upplever du att funktionerna i öppen programvara motsvarar de behov som ni har i organisationen, rent funktionsmässigt alltså. Det är klart att det, ja...
104	M: Alltså det är jättesvårt, det är väl egentligen kanske svår fråga... Finns det något behov i organisationen som inte går att uppfylla med öppen programvara? Så är svaret: Ja. Men alltså om man ska köpa en viss... OK vi ska ha ett sånt här verktyg; räcker då i allmänhet open source funktionaliteten? Alltså om vi ska ha en Office-svit till exempel...
105	Jakob/Jonas: Mm
106	M: Om man tar det som rent generellt exempel...

107	Jonas: Mm
108	M: Ja så e ju svaret... ja... det är ju jättesvårt, det kanske inte är nån bra fråga, men svaret är nog Ja, generellt alltså. På det här med... där det FINNS ett verktyg så, eller det finns en programvara som är öppen...
109	Jakob: Mm
110	M: Så är det ju i stort sett så att dom alltid... alltså det går alltid att hitta nåt som e bra nog skulle jag säga. Sen har vi ju en del... men så finns det ju en del områden där det kanske inte finns nåt, eller ja, då kanske svaret är nej eller det finns inget alls...
111	Jonas: Nä...
112	M: Men det är nog... det blir ju värre...
113	Jonas: Vi skulle kunna kanske ta, vad säger du om att ta Officepaketet som en standard genom frågorna, så vi har något fast att jämföra med hela tiden så blir det lättare för dig också.
114	M: Ja det kanske vara bra ja, det beror lite på vad man är ute efter om man pratar system eller klientprogramvara eller så men... vi kan pröva hur det funkar!
115	Jakob: Det blir lite mer konkret på nåt sätt...
116	M: Om vi tar Office som ett konkret exempel...
117	Jakob: Mm...
118	M: Eh, så... som ligger mig ganska varmt om hjärtat också... så... alltså jag är ingen sån Microsoft-basher på det viset alltså, jag försöker inte att vara religiös så...
119	Jonas: Hehe
120	M: Men eh, så är svaret JA, det är inga problem, däremot så är det ju inget trivialt projekt kanske att gå över. Alltså det finns ju en del personer här som är kanske lite mer rabiata och lite mer naiva och sånthär men "Fan, varför bara slänger vi inte bara ut dom här vi har liksom tusentals licenser, kan vi inte bara kasta dom och så kör vi OpenOffice imorgon".
121	Jonas: Mm
122	M: Men då måste man ha klart för liksom det här, så att alla vet, vi kör faktiskt verksamhetskritiska snurror och processer som är baserade på Excel.
123	Jonas: Mm
124	M: Och problemet är med kalkylblad för att ta ett konkret exempel, att dom är ofta väldigt fragila alltså väldigt bräckliga grejer, man vet inte vad som är... alltså det går inte att se var all funktionaliteten sitter för den e gömd bakom olika checkboxar och...
125	Jonas: Mm...
126	M: Olika formler och olika makron och så vidare, det är jättesvårt. Det är ett kvalitetsproblem i sig med... alltså kalkylblad är fantastiska men dom är egentligen inte så bra att bygga processfunktionalitet i. Men det är ju nånting som många gör i och för sig, alltså, nästan alla, eller väldigt många företag är ju rätt beroende av Excel. Men det gör att det skulle vara långt ifrån en trivial sak att ersätta Excel och även access då med till exempel OpenOffice Base och Calc eller vad det heter.
127	Jonas: Mm, OK.
128	M: Men om man hade börjat från scratch, om vi startar ett företag idag så är det inget snack om saken att det är klart att man skulle kört med OpenOffice, om vi tar det som exempel så att säga.
129	Jonas: Ja OK.
130	M: Hade jag varit IT-chef på en start-up nu, så skulle jag inte lagt en spänn på... jag skulle egentligen inte lagt några pen... det beror lite grann på vilken, eh alltså, demografin på de anställda men jag skulle kanske inte ens kört Windows, det beror lite grann på. Men jag skulle i vart fall inte kört Office, alltså Microsoft Office-paketet, det skulle jag absolut inte gjort.
131	Jonas: Nä...

132	M: Men samtidigt så kommer vi att säkert sitta med det om, ja vi gör säkert det om 3 år här också, kanske inte om 5år men om 3 år ser jag att vi fortfarande kör Microsoft.
133	Jonas: Om vi nu säger, om jag har förstått det rätt att, OpenOffice, det öppna programvara det kan då motsvara behoven, tycker du då att funktionerna som motsvarar varandra då uppfyller samma behov eller samma resultat.
134	M: Det e väl... jag förstår inte riktigt frågan tror jag...
135	Jonas: Eh, till exempel att du kanske kan, funktionen finns att du kan skapa tabeller i både OpenOffice och i Word...
136	M: Jaha du menar att dom är lika bra...?
137	Jonas: Ja precis.
138	M: Ja, alltså det... Ja. Det finns några sådana saker som är lite obekväma i OpenOffice men det finns ju lika många i Office så att.
139	Jonas: Ja, ja...
140	M: Alltså jag... hemma kör jag bara OpenOffice och, så jag har stor erfarenhet av det och på jobbet så märker jag att jag kör både och, alltså ja tänker... Jag öppnar nästan grejer oftare i OpenOffice än jag gör i... fastän jag har då Microsoft Office då på skrivbordet.
141	Jonas: Jasså...
142	M: Av... så uppenbarligen... Jag märker också det när jag supporterar, alltså jag har ingen sån roll riktigt på [Företagsnamn], så egentligen så är klientprogramvara inte riktigt vad jag håller på med men det är ändå ett bra exempel.
143	Jakob/Jonas: Mm
144	M: Men i min privat... som privatpersoner så supporterar jag ett 20-tal släktingar och vänner och när jag sätter upp min version till dom så säger jag "Ni behöver inte betala 1500 spänn eller vad det nu är för Office-paketet, ni kan köra det här". Och det är bara en som har klagat, dom har inte ens märkt att dom har gjort övergången. Och dom är verk... dom är inga... alltså det är helt vanligt folk liksom.
145	Jakob: Hur upplever du att öppen programvara har möjlighet att samverka med andra program?
146	M: Ja den är ju bättre i allmänhet alltså.
147	Jakob: Eh, kan du utveckla det lite grann kanske?
148	M: Ja om vi tar office-sviten i sammanhang så är det ju mycket... alltså senast idag så fick jag Office Publisher-filer som vi då, som vi då inte får installerat så dom kunde jag inte öppna. Det händer ju inte med... får jag det i ett öppet format så kan jag då ta ner programmet och öppna direkt ju.
149	Jakob/Jonas: Mm
150	M: Och ett annat exempel i office-svitens sammanhang är ju att, nu kan ju i och för sig vad heter det... Microsoft generera pdf-filer enkelt från... i... den senaste versionen av då Office och ja kanske då den förra. Men det har ju varit jätte enkelt i OpenOffice i flera år. Det har ju egentligen varit ett killer-argument för just dom i den privata sfären, att folk säger att "Wow! Det här e ju jättebra, det är en knapp i OpenOffice för att generera pdf". Det är ju verkligen en toppengrej.
151	Jonas: Mm
152	M: Eh, och... alltså källformaten, ja om vi nu tar det, vi kör med Office-sviten va?
153	Jakob: Ja
154	M: Källformaten är ju lättillgängliga och det är ju sånt som är viktigt för oss liksom, för det är inte viktigt för användarna.
155	Jonas: Nää...
156	M: Men källformaten för Office, nu vet jag inte hur det ser ut för Office 2007, men dom andra har varit vedervärdiga i tidigare versioner alltså. Eh, och dom är till och med binära. Och man, sen har man ju ovanpå det alltså Works för Windows. Man stöter på ibland där man inte ens kan öppna filer från andra sviter.

157	Jonas: Vad känner du om säkerheten i... Ett exempel då att obehöriga inte kan få tillgång till data du vill skydda, när det gäller öppen programvara.
158	M: Om vi nu ska använda... där kanske inte office-sviten är ett bra exempel utan där kanske vi ska vidga oss. Så är det ju... alltså det finns ju jättestora problem med både sluten och öppen programvara. Många idylliserar ibland den öppna programvara och säger att den är fullständigt oproblematiske. Men den som följer vad som händer... alltså hur många patchar och säkerhetsproblem som... problemet är ju att dom som slår, slår ju mest mot Microsoft-programvara för att det är det som ger störst utdelning.
159	Jonas: Mm
160	M: Men jag menar vi vet ju inte, det kan ju finnas... Jag kan ligga vaken på nätterna ibland och tänka på att vi vet ju inte vad som sker med Mac:ar och Linux-användarna för att det kan ju finnas massor med root-kit i princip som ingen... alltså som just dom som verkligen vill göra nånting i tysthet kan göra detta hålet så att säga.
161	Jakob/Jonas: Mm
162	M: Och där e ju ofta attityden till säkerhet... e ju ofta bristfyllda med användare även om det är ju rätt så... även om dom egentligen är duktigare än den genomsnittliga Windows-användaren.
163	Jonas: Mm, Mm OK.
164	M: Serversidan däremot på systemsidan så... så skulle jag kanske däremot vilja luta åt att jag känner mig tryggare med öppen källkod. Alltså jag känner mig tryggare med Apache än Microsoft Information... vad heter det?
165	Jakob: IIS
166	M: Ja
167	Jonas: Mm
168	M: Men för vilket program som helst gäller ju att det räcker med ett hål för att det ska vara osäkert så att... därför e ju egentligen all programvara osäker.
169	Jakob: Inget är perfekt så att säga?
170	M: Nä
171	Jakob: Om vi skulle ta och flytta fokus lite från funktionalitet och gå över till tillförlitlighet. Har du någon uppfattning om frekvensen av avbrott i programvara på grund av fel i den, så att säga, när det gäller öppen programvara?
172	M: Alltså det... det här är också ett sånt område där man kan... Om vi börjar med Office-sviten som inte är nåt bra exempel egentligen, där har jag nog lite fler gånger... eller där kraschar nästan faktiskt aldrig Microsoft men det händer ibland att OpenOffice när det startar igen gör en recovery på filerna, men alltid utan problem alltså. Men den är något mer instabil. Men om vi då vidgar oss då, vyn till att problem eller alla väsentliga programmen så är ju stabiliteten hos dom öppna programmen... har ju länge haft ett mycket bättre track-record. Nu börjar ju faktiskt... nu börjar det ju komma ikapp på Microsoft-sidan också, vi har ju överlag mycket mycket... så är tillgängligheten hög skulle jag vilja säga vad man än kör om man sätter upp det rätt, det handlar ju mycket mer om organisationen än om programvara kanske.
173	Jakob: Ja det gör väl det... Men skulle du anse... skulle du känna dig bekväm med att köra öppen programvara här med just med tanke på den frekvens av avbrott som...
174	M: Ja, ja det gör vi ju! Och det gör ju massor av andra organisationer också. Jag menar, 49,5% av alla web-servers e Apache och... Det finns ju ingen som får sparken för att man föreslår Apache som webserver, för att ta ett exempel.
175	Jakob: Nä, nä...
176	Jonas: Det är intressant där du sa om också att OpenOffice kör nån recovery där, när det kraschar. Känner du att en öppen programvaras förmåga att återhämta sig efter en krasch är bar eller dålig eller...? Att kunna fortsätta fungera trots att fel uppstår...
177	M: Alltså det är ju bara känslor, man skulle behöva antingen titta på konkreta program eller en massa program och jämföra överlag. Men...
178	Jonas: Känslan är ändå...
179	M: Känslan är att, den nog är nästan ja... det är i vart fall inte något minus för öppen programvara.

180	Jonas: Om man tittar på att det väl skulle krascha, hur... hur... vad e din uppfattning då hur mycket resurser och tid och kraft det krävs för att återställa förlorad data jämfört med nåt annat?
181	M: Det är svårt att generalisera... alltså när det går... det kan ha gått snett på så många olika sätt.
182	Jonas: Mm
183	Jakob: Då skall vi flytta fokus återigen en gång till och gå över till användbarhet. Anser du att det är... har du någon uppfattning om att det är enkelt eller svårt att förstå den logiska strukturen i öppen programvara? Så att säga, skiljer den sig nånting eh...?
184	M: Där har jag en ganska bestämd uppfattning att det kan va lite marigt men det kan också va hopplöst att, framförallt när man ska köpa in slutna programvara. Så... en öppen programvara är ju definitionsmässigt mycket mer tillgänglig, så att man kan se på den. När man går till leverantörer av stora system om vi tar det. Alltså ERP-system och sånt till exempel. Så när du kommer till deras web-siter så är det liksom bara oceaner av ord liksom och man förstår inte ens vad dom gör. Det kan ringa en säljare till en och så "Jag ringer från XYZ och jag ska sälja bla bla bla" och så undertiden så går man in på deras hemsida och så bara blir man jätteledsen för att... vad e det ni gör för nånting? Hur e det uppbyggt, hur fungerar det grovt? Liksom...
185	Jakob/Jonas: Mm
186	M: Och det är ju... När det gäller öppen källkod så kan man gå in och se hur installerar man det rent konkret. Då ser man ju också -OK, där är en sån applikationsserver där, och där e den och den modulen och det funkar så liksom.
187	Jakob: Mm. Innebär detta då, en liten följdfråga här kan man väl säga att, upplever du det lättare att lära sig använda öppen programvara, är det en effekt av det då?
188	M: Ja det e det ju faktiskt, just för att det är tillgängligare.
189	Jonas: Eh...
190	M: Alltså, framförallt, nu får man ju skilja så att, alltså IFS är ju öppen för oss... som det är möjligt att köra den utan att du har begränsningar att du inte kan spara och så vidare. Det här med kostnaden för programvara e ju i allmänhet en liten del av att ta in en programvara. Samtidigt så, med tanke på, som jag skulle säg för att föregripa kanske alla era andra frågor e att kvalitén i allmänhet är väl så bra. Eller att de kanske har olika problem, men dom har i vart fall problem. Alltså både den öppna och slutna programvaran har problem. Så det är det att... det faktum att det bara är att mer och mer köra, man kan sätta upp 8 maskiner och pröva på olika sätt och man kan labba och man kan låta någon annan testa utan att det är en massa jox med licenser. Och utan att man måste motivera den investeringen och att det skall vara fakturor hit och dit. Det är en friktionskostnad som man slipper liksom med öppen programvara. Vad var nu frågan förresten där?
191	Jakob: Jooo, den senaste frågan var väl egentligen ifall det va... men det anser jag nog att du svarat på, ifall det är lätt att lära sig använda öppen programvara?
192	M: Ja, jag råka komma in på det... just... ja den är mycket... den är lättillgänglig alltså. Och dokumentationen är ju... alltså för dom... om man går in på sourceforge så finns det, jag vet inte, en halv miljon projekt eller nåt sånt, där är en massa dåligt dokumenterad programvara. Men det är ju inte den man använder, alltså dom stora programmen man använder vilket är ganska många, dom är ju ofta väldigt väl dokumenterade. Och framförallt är dess dokumentation tillgänglig. Det finns... vill man kan man betala för böcker som är jättebra, det finns hur många... till exempel O'Reilly har ju, det är bara ett förlag som är verksamt inom den domänen och dom har ju hur mycket litteratur som helst, om allt från sendmail till Mono eller vad man nu är intresserad av liksom.
193	Jonas: Om vi säger att du har en specifik uppgift du skall utföra, är det lättare att utföra den då, eller svårare, i öppen programvara eller slutna?
194	M: Alltså om jag har den slutna programvaran, så är det kanske väl så enkelt. Men om det är en... om det är en... om uppgiften... om ni ger mig vilken uppgift som helst, alltså om det är mängden av alla tänkbara uppgifter jag får en dag på jobbet så är det mycket mycket enklare att lösa den med öppen programvara för att jag kanske måste använda ny typ av programvara som jag inte har.
195	Jakob: Om vi går in på, så att säga, attraktivitet har vi kallat det, det vill säga det som syns på skärmen, ifall öppen programvaras förmåga att tilltala dig, rent estetiskt.
196	M: Det varierar ju väldigt alltså men, men om man ska... nu är ju jag inte en person som faller för det estetiska. Men faktum är att ekonomin i den... alltså den öppna ekonomin kring öppen programvara gör ju att det finns egentligen väl så stora möjligheter att... att... jag menar att du... det finns ju... möjligheterna att personalisera exempelvis en UNIX-desktop är ju mycket mycket större. Alltså om du... om man tittar på... alltså det finns folk som e liksom... som... showcasear sin desktop på olika webbsidor så det finns många många fler estetiskt tilltalande UNIX-desktops än det finns Windows-desktops. Men om man sätter programvaran out-of-the-box så är det klart att Microsoft Office

	ser kanske lite mer flashigt ut än OpenOffice, men det är inte så det spelar någon roll alltså.
197	Jonas: OK...
198	M: Och även dom med skins och sånt... där drivs ju även dom... alltså hela den här skins-grejjen som... alltså jag är ju inte en person som skinnar om mina programvaror men jag vet ju att alla... att dom flesta då framförallt yngre människor gör det hela tiden. Den ekonomin är redan öppen även för slutna program för att det, det går inte ju att ta betalt för det utan...
199	Jonas: Nää
200	Jakob: Men så att säga standardutförande...
201	M: Standardutförande kan ju, e ju i allmänhet lite mer välpolerat för slutna programvara med undantag skulle jag vilja säga egentligen av window-managers.
202	Jakob: Mm...
203	M: Som är ofta väl så tjugiga. Men om du liksom tar en konkret applikation så kanske det inte är... ser lika flashigt... alltså GIMP ser kanske inte lika attraktivt ut som... nån av dom här... slutna grafikbehandlingsprogrammen. Även OpenOffice är inte riktigt lika välpolerat som Microsoft.
204	Jonas: OK. Nu kommer vi in på lite mer inre kvalitet i programvara. Hur upplever du bearbetningstider, svars- och bearbetningstider, vid exekvering i då öppen programvara?
205	M: Ja, det har man ju naturligtvis problem med ganska ofta men då har man ju... då e det ju kanske inte så att man har möjlighet att jämföra utan man sitter ju med det man har. Alltså det har ju... jag glömde säg det att när det gäller... vad har vi för slutna programvara? Vi har ju en del Oracle-databaser.
206	Jonas: Mm?
207	M: Och vi har egentligen kanske... i stort sett så har vi... använder vi fler databaser, vi använder Access, helst inte men... men det finns en del grejer som är verksamhetskritiska som går på Access. Sen har vi Oracle, framförallt på ITS, kör på Oracle och så resten på My-SQL i stort sett. Och... alltså där finns ju egentligen inga... man kan inte tala generellt om skillnader...
208	Jonas: Nähe?
209	M: Om man går ner på desktoppen igen så tar det ju tid att starta Windows, det tar ju en tid att starta en typisk Linux-desktop också faktiskt, ungefär lika lång tid.
210	Jonas: Mm...
211	M: Men dom är... om man går ner desk... alltså... Man kan väl säga såhär kanske egentligen att typisk öppen klientprogramvara kan väl vara lite snabbare just för att den är lite mindre flashig och så vidare. Men däremot... så kan det väl ibland vara så att en del serverprogramvara typ databaser och sånt där har dom slutna upphovsmännen liksom lagt väldigt mycket pengar för att... alltså... dom lägger väldigt mycket pengar på att det ska vara väldigt väldigt snabbt och det är därför dom kan ta betalt också. Alltså då betalar man kanske för extrem tillgänglighet och comformance, men där kommer ju i allmänhet open source verktygen ikapp. Sen en programvara har ju den fördelen att den inte... när den väl e fixad så går den ju inte isönder igen. Nu vet ni att det är inte riktigt sant men i stort sett... Så det blir kanske mindre och mindre nånting man kan leva på...
212	Jonas: Mm, hur känner du då, om vi tittar på då; dom resurser som öppen programvara utnyttjar i form av ja, hårdvara, personal, e det nånting som det går att ha en uppfattning om?
213	M: Hm, nä... hårdvara det är rätt så hugget som stucket, det kanske... ja det kanske skiljer nån faktor 5 % men det är ju som inom bred marginal... ja det e klart det kan ju vara upp och ner också ju. Personal e ju inte heller nån skillnad skulle jag vilja säg. Alltså för oss som utvecklar en hel del av vår egen programvara och integrerar då öppna program som nu drar en del personal men det hade ju det gjort även om vi hade använt... alltså då kanske det ofta hade varit så att vi hade tagit in nån konsult men det kanske... men det blir ju inte mer eller mindre... det blir ju inte mer eller mindre tid alltså. Om vi tar som exempel, vi hade ju lite grann som... om vi går tillbaks till Office-sviten, så bortsett från de här problemen med att migrera från det vi har idag till det öppna... och så med min erfarenhet från den privata sfären så skulle jag bedöma att för användarnas del skulle det inte varit något problem att byta, eller litet problem att byta. Underhåll skulle inte varit besvärligare för helpdesken, alltså dom skulle få frågor om... dom som frågar idag om olika grejer om vad som helst skulle... det skulle dom få med OpenOffice också. Jag kan inte tänka mig att det skulle vara någon skillnad, alltså vilka helpdesk personer man behöver.
214	Jakob: Ifall vi skulle då gå vidare till underhållsmässighet, eller maintainability som kanske är en mer naturlig term...

215	M: Jag har en rolig historia här, det har väl inte riktigt med det att göra men. Alltså det illustrerar kraft i öppen programvara. Vi hade problem med en utav våra routers för nåt år sen. Och vi hade väldigt svårt att...
216	Jonas: Som ni säljer då?
217	M: Nä nä, alltså en annan... här i huset för eget nätverk och koppling till intranet.
218	Jonas: Mm
219	M: Och vi hade då, visade sig, betalt för för låg supportnivå sen... alltså då när den krånglar så behöver man ju jättemycket support även på natten och tydligen så hade vi missat det i nåt avtal och så vidare. Då va det några av dom killarna här som kan det på systemavdelningen som satte upp en router i en gammal desktop-låda i stort sett så vi kunde köra på, alltså den var ju inte lika snabb naturligtvis för det va ju ingen dedikerad hårdvara. Det gjorde vi... det gjorde dom på bara någon timme så va vi uppe och körde liksom.
220	Jonas: heh, tufft.
221	M: Jag kom och tänka på det... ja vi prata just nu om... ja det kan kanske visa lite grann på... på många av dom kvaliteterna som man uppnår alltså.
222	Jonas: Mm
223	M: [Ohörbart]
224	Jakob: Har du någon uppfattning om det arbete som krävs för att identifiera brister och felkällor i öppen programvara? Eh hur skall man kunna få ut nåt mer konkret... kan du hjälpa mig här lite Jonas?
225	Jonas: eh...
226	M: Alltså felsökning och så?
227	Jakob: Ja precis, precis! Ja, arbetet med felsökning helt enkelt.
228	Jonas: Och kunna hitta brister för att sen kunna eventuellt uppgradera.
229	M: Ja det skulle jag säg att det är också... alltså alla mina svar är väldigt subjektiva. Jag försöker ju grunda dom i nånting men jag skulle ju säga att dom är... alltså en bias som finns i detta är att jag i väsentligen inte är någon... alltså även om jag inte är någon Microsoft-basher och jag kör på primära desktop Windows XP, även om jag har en Linux-maskin också, på jobbet alltså. Men det är egentligen bara OS:et jag använder för det alltså, och sen utför jag har i praktiken väldigt mycket jobb på dels på... i... i UNIX-skalet, som jag kör på Windows-maskinen av praktiska skäl men på min, på min UNIX-maskin. Och till exempel i My-SQL så... vad ska jag nu komma in på... Jo, vad gäller programutveckling och sånt som är det hållet jag kommer ifrån, alltså det jag börjat med, programmering och så. Så har jag egentligen inte använt Visual Studio och .NET och såna här teknologier. Så att jag är... kan vi säga ingen Microsoft-person. Så att jag e... jag kan ju sämre vilka verktyg och redskaps som behövs egentligen, men vad man kan säg är ju att för öppen källkod och UNIX-sidan så finns det ju i allmänhet massor med redskap och felsökningsverktyg och debuggar och loggningar o allt vad det... dom kan vara ganska svårtillgängliga ibland men att göra den typen utav avancerad felsökning är aldrig enkel att göra vad man än använder för miljö. Och sen är det ju det att källkoden är öppen, definitionsmässigt för öppen källkod, så behöver man så kan man ju titta där också, så den fördelen har man ju. Och man kan instrumentera programvaran och så vidare, men ni ska veta det att... alltså det är nåt man i praktiken aldrig gör, men möjligheten finns ju.
230	Jakob: Nästa fråga här... Ja, jag vet inte om den svarar på sig själv frågan här... men... Hur anser du att möjligheterna är att modifiera eller uppgradera öppen programvara?
231	M: Ja, den är ju naturligtvis inte mindre, för den är ju större. Där skulle man kanske ha nåt konkret exempel. Modifiera är ju i vart fall enklare. För ofta... möjligheten att vem som... dels så går den ju i sig att modifiera sen är ju möjligheterna att... alltså typiskt möjligheterna att parameter-styra konfigurationen eller att programmera i nån sorts script-språk och så vidare, den är ju ofta väl så stor i de öppna redskapen som i... som i... Man kan ta Apache som exempel, den är ju superflexibel alltså men... men... möjligheterna att [Ohörbart] i vilket programmeringsspråk man vill i stort sett och en massa möjligheter att konfigurera och...
232	Jonas: Mm... Om vi säger att du har till exempel har gjort en sån här modifikation av nåt slag med din programvara, vad anser du då med möjligheterna att kunna testa den programvaran så att säga? Liksom att modifikationen har uppnått det som... eller så att det inte har blivit några fel eller... ja.
233	M: Alltså den frågan finns ju egentligen inte för slutna programvara, om man inte kan modifiera. Om man gör modifikationerna i nån slags anpassningslager som finns i programvaran, det finns scripting exempelvis eller makron eller vad det kan vara. Så är det i vart fall inte svårare att göra det för öppen programvara.

234	Jonas: OK. Kan det vara så till och med att uppdateringar som kommer till programvaran kan få en önskad effekt på programvaran?
235	M: Ja det kan det va, men det kan det verkligen ha även för... alltså det har du även för slutna programvara.
236	Jonas: Mm
237	M: Det har blivit bättre men det är ju... där är ett stort... det där är ett potentiellt stort... det är ganska sällan som saker och ting går sönder på grund av uppdateringar. Men när det händer så... alltså egentligen så vill man ju ha... jag menar större företag har ju... eller... jag inbillar mig nog att större företag har det men jag är inte så säker på det... att man liksom innan man tar in en uppgradering så testas man den... alltså det kräver ju oerhört... tänk om man testas alla Microsoft-updates som kommer... det är ju en jättegrej. Så det gör man ju inte, så i praktiken så... när det gäller patcher... alltså... uppgraderingar är en sak och uppdateringar är en sak. Jag menar uppdateringar så håller man ju... så ber man en bön och kör in den...
238	Jakob/Jo/M: skratt...
239	M: Uppgraderingar e ju samma... e ju nånting man gör kontrollerat men det är ju ungefär... det gör man ju kontrollerat för köpe-programvara och det är kontrollerat för öppen programvara. I allmänhet går det ju bra, ibland går det dåligt men det är inte svårare för någon av dom ju.
240	Jakob: Ja, om vi skulle ta och diskutera lite flyttbarhet, eller portability som är den engelska termen. Anser du att öppen programvara är möjlig att anpassa till er verksamhet här? Har du någon uppfattning om just anpassbarheten?
241	M: Vad det inte den vi... pratade om egentligen? Ja det är samma svar så att säga. [Ohörbart]
242	Jonas: Sen har vi då dom ansträngningar som behövs göras för just installation av öppen programvara, alltså just själva momentet att installera. Hur upplever du dem?
243	M: Återigen, då har jag en lite mindre erfarenhet av just Microsoft... Om man kör Debian till exempel som vi gör typiskt för Linux-programvara så är installationen i [Ohörbart] enkel. Där ligger ju egentligen öppen programvara före skulle jag vilja säga. Sen har vi några större programvarasystem, vi hade ju installerat dom också i paket... det kan ju va... ja det är ju i vart fall inte svårare i allmänhet, det är faktiskt bättre.
244	Jonas: Mm, OK.
245	Jakob: Om du skulle tänka dig... ifall du försöker föreställa dig en programvara som ni kör idag, som t.ex. Office...
246	M: Mm
247	Jakob: Om du skulle tänka dig att ni skulle bytt det mot OpenOffice kanske, vilken... har du någon uppfattning det skulle kräva just i och med själva bytet? Övergången där...
248	M: Ja, det skulle ju va ett ganska stort... Där är det ganska lätt att säga vad som skulle va problem, och det skulle ju va... Det är två problem egentligen, det ena har jag inte nämnt. Det ena är att vi är ganska beroende av hur vi kör... Vi använder Exchange idag och plus Outlook för mötesbokningar, och det funkar ju väldigt väl men det skulle ju kräva nån initierande av motsvarande lösning utav Exchange och Outlook, men det finns ju såna idag. Så det är ju rätt litet... det skulle egentligen va... det är ju inget... överkomligt arbete. Sen skulle vi behöva identifiera alla tjänster och som finns långt bakåt som är kritiska, och gå igenom dom. Men det är nånting vi hade haft glädje av att göra ändå.
249	Jakob/Jo/M: skratt...
250	M: Och det hade ju varit ett arbete... men det hade ju... och sen så hade vi kanske behövt nån... sen behövs ju nåt genomförande av utbildning och konsultning och så vidare. Men det e ju inget... alltså det skulle... det skulle gå att planera ett sånt projekt på ett kontrollerat sätt och det skulle va överkomligt arbete.
251	Jonas: Men det skulle vara krävande?
252	M: Ja det skulle ju va en del arbete men det är jättesvårt att säga hur mycket det skulle va...
253	Jonas: Mm... Nä... Där va vi nästan...
254	Jakob: Ja nu har vi bränt vårt frågebatteri.

8.4.3 Vårdenhet

1	VÅRD: Vad heter det, vad ska uppsatsen...
2	Jakob: Den ska handla... Titeln är ju lite sådär en så länge, men den ska i alla fall handla om öppen programvara
3	VÅRD: Ja.
4	Jakob: Och... det är ju såhär att det finns en hel del pengar i licensavgiften att tjäna in på att använda sig av öppen programvara. Men ändå så är det väldigt få som... företag och organisationer som använder sig av det.
5	VÅRD: Precis
6	Jakob: Och då håller väl vi på att luska lite grann varför det kan vara så. Då tänker vi oss att det kan vara något så som mjukvarukvalitet.
7	VÅRD: Ja
8	Jakob: Alltså kvaliteten i programvaran som skulle kunna vara en av anledningarna så väljer vi att titta lite närmare på just uppfattningar om mjukvarukvalitet i verksamheter
9	VÅRD: Ja
10	Jakob: Ja, så vi hoppas väl...
11	VÅRD: Ja, jag nämnde ju för dig då när du ringde att jag hade då precis någon månad tidigare varit på ett seminarium i Stockholm som just handlade om... ja lite om standardisering, men också mycket om öppna system
12	Jonas: Mm
13	VÅRD: Och precis varför ingen vågar gå in i det mer än vad polisen gjort det tydligen, som Rikspolisstyrelsen
14	Jonas: Mm
15	VÅRD: Det känner ni kanske till?
16	Jonas: Ja just det
17	Jakob: Mm
18	Jonas: Vi läste det nu just nyligen
19	VÅRD: Ja. Det var något öppet med det
20	Jakob: Mm
21	VÅRD: Så att det är nog många som... Man är rädd för att göra
22	Jakob: Mm
23	VÅRD: Man vet inte riktigt var det innebär
24	Jonas: Mm
25	VÅRD: Det är bättre att köra med standarden, Microsoft, eller något sådant så är inne i fällan ju
26	Jonas: Mm precis
27	VÅRD: Som Region Skåne har gjort ju...
28	Jakob: Ja precis ja... Skulle du kort bara kunna beskriva vad du gör
29	VÅRD: Vad jag gör?

30	Jakob: Ja precis...
31	VÅRD: Ehh jag är då IT chef här på [företagsnamn] så det är en förvaltning av av... ja 12-13 förvaltningar i Region Skåne, varje storsjukhus är en förvaltning ju. Lund, Malmö och så vidare. Men [företagsnamn], vi är ju, täcker ju denna regionen så vi har ju vår egen förvaltning på [företagsnamn]
32	Jonas: Mm
33	VÅRD: Framför allt
34	Jonas: Hur många anställda är det som tillhör...
35	VÅRD: Vi är ju 850 anställda och sen är det en IT enhet på 7 personer. Eftersom vi då inte är något storsjukhus så har vi egen IT avdelning. Det har inte storsjukhuset på det sätter utan vi har egna system, det är ju lite annorlunda verksamhet
36	Jakob: Mm
37	VÅRD: Så jag är ansvarig för att systemen som finns här. Det är ju dels då gemensamma system som vi har besöksregistrering. Det ser ju likadant ut där, var man än är. Journalsystem har vi då eget.
38	Jakob: Mm.
39	VÅRD: Då följer vi inte regionens stora journalsystem.
40	Jonas: Mm ok
41	VÅRD: Så att därför har vi lite mer resurser än, till förhållande till vår storlek.
42	Jonas: Mm
43	VÅRD: Så det gäller att följa upp och hitta nya områden, nya IT områden som... Det vi jobbar mycket med nu det är egentligen mer det här med analyser och stoppar sådana här analys modeller i olika sammanhang, uppföljningar och...
44	Jakob: Mm
45	VÅRD: Jag har den bakgrunden, jag har ju bara varit här 1,5 ja nästan snart 2 år nu faktiskt... tiden går
46	Jonas: Mm
47	VÅRD: Jag kommer från VM data. Jag har varit konsult där i 10 år så jag kommer ju från andra tidigare... så jag vet lite
48	Jakob: Mm
49	VÅRD: Nu har vi rätt lite konsulter inne. Vi har ju eftersom vi har egna resurser
50	Jonas: Mm
51	VÅRD: Så... och det var ju diskussioner på VM data då också, det här med öppna system. Om kunden också skulle införa öppna system och så
52	Jakob: Mm
53	VÅRD: Det är få, fall där man sett...
54	Jakob, Jonas: Mm
55	Jakob: Men diskussionen har förekommit...
56	VÅRD: Jaja
57	Jakob: Varit återkommande kanske...
58	VÅRD: Det var ju särskilt nu när regionen gick in och skulle förlänga avtalet med Microsoft, om man skulle göra det till... det här med Windows plattform... ehh men man valde ändå för att vara på säkra sidan

59	Jonas: Mm
60	VÅRD: Det är ju indelat så, det ska jag säga också... Jag är ju alltså IT chef för en förvaltning och sen är det då IT chefer på de andra förvaltningarna eller IT samordnare heter det kanske... Sedan så... så är det en övergripande IT enhet på regionen och de ansvarar för sådana här gemensamma plattformar. Så det är de som gör, tar... I samråd med oss tar beslut för det... Nu går vi vidare här t.ex. på nya... Vi ska ju införa ett dokument hantering system, sådana system som är gemensamma för alla.
61	Jonas: Mm ok. Har du något inflytande i valet av någon programvara här för er...
62	VÅRD: Lokalt
63	Jonas: Mm
64	VÅRD: Ja, Det, det har jag ju, men det är ju lite begränsat så... att man ska ju följa dom, den IT-strategin som regionen har ju
65	Jonas: Mm
66	VÅRD: Men vi... För vårt journalsystem så har vi ju en helt egen plattform, då har vi Lotus Domino Notes
67	Jakob: Mm
68	VÅRD: Som inte är så vanligt, det är väl lite utdöende, så vi har en egen plattform där. Där följer vi inte Microsoft, det är ju... värsta konkurrenten för Microsoft
69	Jonas: Mm
70	VÅRD: Var en gång... Det är IBM som satsade på Notes ju.
71	Jonas: Mm
72	Jakob: Ehh, om vi gör såhär, vi har ett par... ett gäng frågor här, ett batteri liksom.
73	VÅRD: Ja
74	Jakob: Ehh, dom är väl rätt så specifika, de flesta av dom.
75	VÅRD: Ja
76	Jakob: Så sätter vi sedan ihop en helhet, av det intryck vi har fått här.
77	VÅRD: Javisst
78	Jakob: Ehh och när vi... efter intervjun... kan säga det kort bara, så kommer vi sätta oss ned och transkribera detta, och...
79	VÅRD: Ja
80	Jakob: eventuellt att ni tar ut meningar som, ehh, hur brukar vi säga... som kortfattat beskriver det som vi upplever att du menar, ifall det skulle vara så att det svävar iväg och blir alldeles för...
81	VÅRD: Ja just det
82	Jakob: Ehh och den här tolkningen eller vad man ska säga, eller analysdelen... ehh skulle vi kunna maila till dig om du vill läsa igenom den.
83	VÅRD: Ja det kan jag göra
84	Jakob: bara för att kontrollera att... att du står för det som vi ska använda oss av senare så att säga
85	VÅRD: Ja
86	Jakob: Och så är det ju även så att, du får tillfälle att vara helt anonym i uppsatsen
87	VÅRD: Jaja

88	Jakob: För att det finns ingen anledning för oss att... liksom beskriva i detalj liksom.
89	VÅRD: Ja
90	Jakob: Så om det känns okej med dom villkoren så vill vi gärna fortsätta med lite frågor här.
91	VÅRD: Ja
92	Jonas: Det är väl en... 20 stycken frågor vi har, ungefär. Där vi hela tiden kommer fråga om din uppfattning på olika saker och ting.
93	VÅRD: Jaja, om jag nu har en uppfattning
94	Jonas: Ja, precis... rörande kvalitét och öppen programvara.
95	VÅRD: Ja
96	Jonas: och är det så att du inte har en uppfattning, så är det helt okej. Det är en del av resultatet.
97	VÅRD: Ja... ja precis.
98	Jonas: och sen så är det så att vi själva har ingen uppfattning om öppen källkod, eller... öppen programvara, om det är bra eller dåligt
99	VÅRD: Nä, just det
100	Jonas: utan de... vi vill bara höra din uppfattning
101	VÅRD: Ja, ja nä men det är väl okej.
102	Jakob: Ehh, om vi börjar här med funktionaliteten i programvaran då...
103	VÅRD: Ja
104	Jakob: Anser du att det finns alternativ till den programvaran i har idag som har samma funktioner, eller har dom funktionerna som ni skulle behöva, i själva organisationen
105	VÅRD: Ja, där finns alternativ ja.
106	Jakob: Mm, rent funktionsmässigt
107	VÅRD: Ehh rent funktionsmässigt ja, till exempel vi har ju då en sådan standard programvara som Office paketet.
108	Jakob: Mm
109	VÅRD: Där finns ju Open-Office som man har diskuterat mycket i regionen om man ska införa... gå in för... och då
110	Jakob: Mm
111	VÅRD: Och dom anser då att, där är mycket funktionalitet i Office... I MS Office som man inte använder va.
112	Jonas: Mm
113	VÅRD: Och där är mindre i det här Open-Office, men den är kanske fullt tillräcklig för det dagliga man använder... så att där finns alternativ
114	Jakob: Mm
115	Jakob: Ehh, sen kan det ju även vara så att även om funktionen finns, ehh... så kan resultatet man får ut av funktionen kanske variera lite grann
116	VÅRD: Ja
117	Jakob: Låt säga att vi har ... ja vad ska vi ta som exempel, kanske... Om vi håller oss till Office då så kanske det finns att infoga en tabell av ett slag och sådär...
118	VÅRD: Ja

119	Jakob: och så finns det ju många olika typer av tabeller, olika detaljerade osv... Anser du även att resultatet av en funktion i öppen programvara skulle kunna lämpa sig för det behov som ni har här?
120	VÅRD: Ja, det är... Ja, det kan det göra. Det som vi haft krux med då är att... att veta om det fungerar mot t.ex. andra... integrationen med journalsystem och sådant. Det är väl det man diskuterat mycket.
121	Jonas: Mm
122	VÅRD: och då kan ju resultatet bli annorlunda om inte integrationen fungerar fullt ut ju
123	Jakob: Mm
124	VÅRD: Så risken finns att det blir ett annat resultat ju.
125	Jonas: Mm, Då har du nästan täckt av vår andra fråga också
126	VÅRD: Jasså
127	Jonas: Tänkte höra lite om hur din uppfattningar om öppen programvaras förmåga att samverka med andra system, men det är väl lite av vad du sa
128	VÅRD: Ja, ja och det, den erfarenheten jag har så kan det vara begränsat då att... att den samverkan kan ju vara mindre med öppen programvara
129	Jonas: Mm
130	Jakob: Mm, tänkte höra om du har någon uppfattning om... Om vi tittar på säkerhetsaspekterna i öppen programvara, ehh... D.v.s... ehh... Finns det några risker att obehöriga får tillgång till data osv.? om du har någon uppfattning om det
131	VÅRD: Ja, alltså jag tror inte att det är en större risk med öppen programvara egentligen för att här på regionen så har vi ju väldigt hög nivå på brandväggen... skyddet så att säga. Så vi har... så jag ser inte det som någon större risk kanske med öppen programvara för att det är inom den här... vår brandvägg så att säga. Och eftersom vi har väldigt kontroll på det va, man har lagt sig väldigt högt, som då har varit till nackdel, är till nackdel i vissa... andra funktioner
132	Jakob: Mm
133	VÅRD: Ehh som vi här på [företagsnamn] har vi ju... vi har ju synskadade, och hörselskadade, framförallt hörselskadade. Även medarbetare, det är väl lite så, nära till hands... Man anställer de som är hörselskadade också eftersom vi jobbar med det. Ehh... Då har vi ju bildtelefoner för de hörselskadade, så att de kan teckna via dator och bildskärm till andra. Vi kallar det... Ni vet brukare, ni vet väl det, vi kallar dom för brukare, vi kallar dom inte patienter ju, det är brukare ju. Så när vi ska ha... de ska sitta och chatta med brukare ute för det är mycket kuratorer och psykologer
134	Jakob: Mm
135	VÅRD: Då kommer de inte igenom brandväggen med sin bildtelefon för det är annan programvara.
136	Jonas: Okej
137	VÅRD: Det är ingen öppen programvara men det är en annan programvara
138	Jonas: Mm
139	VÅRD: Det är lite... Det är då... Eftersom vi lagt det så högt så... så har vi problem där... med sådana här specialfunktioner
140	Jonas, Jakob: Mm
141	VÅRD: Men vi håller på att lösa det men man får gå runt det så att säga med sådana här VPN-tunnlar och sånt
142	Jonas: Ehh, då tänkte vi prata lite om tillförlitligheten hos öppen programvara, hur den uppfattas
143	VÅRD: Ja
144	Jonas: Ehh, har du någon uppfattning om antalet oväntade avbrott i en öppen programvara, exempelvis då jämfört med en annan

145	VÅRD: Ja, ja då har ja lite... alltså jag har en liten erfarenhet av det så det är svårt för mig att veta om det blir mycket avbrott men... Ehh... det kan jag väl tänka mig att den risken för avbrott är lite större kanske
146	Jonas: Mm
147	VÅRD: Ehh, men öppen programvara för de... Det är väl de här också lite med hur supporten fungerar och så
148	Jonas: Mm
149	Jonas: Om vi säger att vi då får ett sådant här oväntat fel i programvaran
150	VÅRD: Mm
151	Jonas: Har du någon uppfattning då om programvarans förmåga att ändå fortsätta fungera, trots ett lite fel eller att allt bara brakar ihop.
152	VÅRD: Näe, det tror jag inte, utan det fungerar... Det kan fungera ändå, fortsätta med ett lite fel ju.
153	Jonas: Mm
154	Jakob: Mm
155	VÅRD: Det är väl tvärt om med då... standard programvara att... ett litet fel kan bli ett stort avbrott ju
156	Jonas: Mm
157	Jakob: Mm, ehh... Låt säga nu att... ehh... Man investerar i ett system med öppen programvara som bas... ehh... Och det här systemet skulle krascha
158	VÅRD: Mm
159	Jakob: Ehh... Det krävs ju alltid vissa resurser för att återställa systemet till ett stadie innan kraschen då.
160	VÅRD: Ja
161	Jakob: Tror du att det finns någon skillnad mellan antalet resurser som krävs för att återställa ehh... ett system baserat på öppenprogramvara respektive slutna programvara
162	VÅRD: Ja, jag tror... Ja det går nog åt mer resurser i öppen programvara, det är väl det som är lite problemet ju, det går nog mer åt resurser än slutna programvara
163	Jakob: Mm, hur va...
164	VÅRD: Där kan man nog ringa in problem lättare, i slutna programvara, lättare och åtgärda... I öppen programvara kan det ju bli svårare att hitta orsaken till felet
165	Jakob: Mm
166	VÅRD: Så jag tror det går åt... det är väl där det... De "tvischar" lite om att, dom som, dom mot eller för... Det går åt mycket mer resurser åt öppen programvara. Det är billigare att köpa in men det är ju dyrare att sen underhålla. Men det kanske är så högt med... att köpa in standard programvara så att man har mycket pengar att hantera... för att kunna klara de där resurserna
167	Jonas: Mm
168	VÅRD: Det kan bli billigare ändå i längden
169	Jakob: Mm, ja precis
170	VÅRD: Om man slipper sådana här support... avgifter... årsavgifter och sådant som... De är ju rätt så... De ligger ju på 20 procent på de kompletta standardprogrammen. Om vi köper in en programvara för 1 miljon så betalar du ju då 200 000 i årsavgift.
171	Jakob: Mm
172	VÅRD: För att kunna få ny version och så va
173	Jonas: Men då får ni det med licenskostnaden

174	VÅRD: Ja då får vi med det, men... Man måste betala in varje år ju, man får inte bryta det ju. Men det slipper man ju med öppen programvara, där är inte den års... årsavgiften
175	Jakob: Är det mer så då att du betalar, när du väl behöver hjälpen då med ...
176	VÅRD: Ja precis, det blir mer löpande räkning
177	Jakob: Mm
178	VÅRD: Lite osäkrare men det är ändå... Om man har mycket pengar kvar i kassan om... tills det händer så kommer vi se om det räcker
179	Jakob: Mm
180	VÅRD: Oftast räcker det kanske
181	Jonas: Mm
182	VÅRD: Och det är ett gissel det här med standardprogramvara, att det är så höga årsavgifter
183	Jakob: Mm
184	VÅRD: Alltså år efter år... rullar på
185	Jakob: Ja, jag kommer inte på några exakta siffror nu men... Jag vet jag läste någon rapport... Ehh ja för svenska staten då skulle kunna tjäna in på bara licensavgifter
186	VÅRD: Ja
187	Jakob: ... hur mycket som helst, men där var kanske inte alternativet viktigt, räknat på, utan där hade man bara sammanställt då... licenskostnaderna för staten så att säga
188	VÅRD: Ja, nu på regionen så köper vi in ehh, programvaror för flera hundra miljoner årligen, och så 20 % på det... det är rätt mycket pengar.
189	Jakob: Mm
190	VÅRD: 15 - 20% procent varierar det med tror jag
191	Jakob: Mm
192	Jonas: Mm, Ehh... Om vi tittar lite på användbarheten
193	VÅRD: Ja
194	Jonas: Ehh... och själva strukturen hur ett program är uppbyggt.
195	VÅRD: Mm
196	Jonas: Tror du det är lättare eller svårare att begripa ett med öppen programvara?
197	VÅRD: Ja du... Nä där har jag inget bra... inget bra svar... jag vet inte riktigt... Det kan ju variera väldigt...
198	Jakob: Det är kanske lite...
199	VÅRD: Ja det är möjligt att det kan vara lättare i öppen programvara
200	Jakob: Jag tänkte ifall man kanske tänker sig en situation för er organisation här, där ni trots allt har kanske har utvärderat ett alternativ och har det i åtanke så är det kanske lite lättare att konkretisera tankarna. Så att säga... Om ni nu har tänkt på Office, att byta det och så vidare liksom... Den logiska strukturen i alternativ programvara där då... som... Du nämnde innan Open-Office
201	VÅRD: Ja
202	Jakob: ... kanske och så vidare
203	VÅRD: Ja det kan ju va... Alltså jag tror inte att det är bättre struktur i öppen programvara på det sättet, näe det tror jag inte. Det kan nog variera.

204	Jonas, Jakob: Mm
205	Jakob: Tror du att, ehh... Blir det då även... Påverkar detta svårigheten med att lära sig... ehh programmet, att lära sig använda det, tror du
206	VÅRD: Näe det tror jag inte, alltså använda det, alltså för en användare att använda det
207	Jakob: Ja precis ja
208	VÅRD: Nä det tror jag inte har någon betydelse
209	Jakob: Så att ehh...
210	VÅRD: Det är fördel... Det är klart att det är en fördel med... likriktningen på standard programvaran, att det ser lika dant ut i alla situationer... Det är lättare, man vet att om man trycker på en viss tangent så blir det en viss funktion ju, det kan variera i öppen programvara, det följer inte med någon sådan standardisering.
211	Jakob: Mm
212	Jonas: Ehh... Anser du att det är tillräckligt lätt att utföra en uppgift i en öppen programvara, om du då sätter det i relation till en slutenprogramvara
213	VÅRD: Vad menar du, en ändring?
214	Jonas: Du har en viss uppgift som du ska utföra
215	VÅRD: Utföra ja...
216	Jonas: i datorn
217	VÅRD: Ja
218	Jonas: och...
219	VÅRD: Om det är lättare...
220	Jonas: Mm, eller svårare
221	VÅRD: Ja, det kan vara svårare i öppen programvara att utföra en uppgift ja
222	Jonas: Mm
223	VÅRD: Därför att... Det kan ju då vara att... Det är svårt att se kopplingar och sammanhang och så...
224	Jakob: Mm, har du någon uppfattning om... något vi kallar attraktivitet i öppen programvara, alltså ifall det är estetiskt tilltalande.
225	VÅRD: Ja, det är ju rätt så betydelsefullt
226	Jakob: Ja för vissa så är det ju det
227	VÅRD: Ja, det kan ja tänka mig att de, det är väl lite det med användarvänlighet, man kan ju bygga upp en viss... En rätt så bra användarvänlighet och då blir det attraktivt ju. Och du kan kanske bygga ihop, du har ju flera öppna programvaror som du kan bygga ihop så får du precis som du vill ha det ju. Det får du inte i standard programvara ju, på det sättet... eller sluten programvara. Där köper du en stor cirkel men du använder kanske bara en liten sektor nere i... om man ser funktionsmässigt.
228	Jakob: Mm
229	VÅRD: Men i öppen programvara, där bygger du ju upp precis vad du vill ha
230	Jakob: Mm
231	VÅRD: Så att det kan ju vara attraktivt ju... Mm
232	Jakob: Okej, om vi då går in på lite mer, det här är väl snarare tekniska begrepp här... produktivitet... ehh... ifall du har någon uppfattning och svars och bearbetningstider för att utföra, alltså exekvera en funktion i öppen

	programvara?
233	VÅRD: jämfört med en sluten?
234	Jakob: Ja, precis ja, om du tänker på något system du har idag och sen så alternativen och hur det skulle se ut
235	VÅRD: Ja det kan bli effektivare med slut... öppen programvara för då gör du bara det du precis behöver göra ju... Jag kan tänka mig det kan vara effektivare
236	Jakob: Mm, en annan grej här som, som även kan spela in är ju vilka resurser det krävs för att köra en viss programvara, i form av hårdvara och personal och material osv., även annan mjukvara som måste kombineras och så... ehh... finns det någon skillnad mellan sluten programvara och öppen programvara där tror du?
237	VÅRD: Ja, vi får se här... Näe, det tror jag inte är någon större skillnad där... det tror jag inte, mellan öppen och sluten.
238	Jonas: Näe, ehh... Om vi tittar lite på hur det är att underhålla en programvara
239	VÅRD: Ja
240	Jonas: Så undrar vi lite, vad din uppfattning är när det gäller att hitta och identifiera fel i en programvara
241	VÅRD: Ja
242	Jonas: Om det väl uppstår fel så ska du identifiera det, hur du upplever det?
243	VÅRD: Jag tror det kan vara lite svårare i öppen programvara att identifiera fel, därför du är mer... det beror på det... hur du själv har byggt upp det, jag menar har du en sluten programvara så har du kanske, från leverantören en vägledning, vad... som symptomen av fel och då kan man lättare ringa in felet kanske... Men det är nog, det är nog svårare i sluten programvara för det är ju unikt ju, mer unikt ju... på den användningen som du just då har ju
244	Jakob: Mm
245	VÅRD: Så där är skillnaderna tror jag.
246	Jonas, Jakob: Mm
247	Jakob: Mm, om vi tittar på, på... modifierbarhet, dvs. möjligheter att förändra och uppgradera osv. Det finns ju också vissa meningsskiljaktigheter här kanske?
248	VÅRD: Ja
249	Jakob: Jag vet inte om du har någon?
250	VÅRD: Ja där är ju också en fördel med öppen programvara, du kan modifiera de, mer ju, än sluten programvara. Där måste du ju ta hänsyn till alla de andra intressenter som kör den programvaran ju.
251	Jakob: Mm
252	VÅRD: Där är det lättare att modifiera de ju
253	Jakob: Mm
254	VÅRD: Det kan ju kosta en hel del ändå och modifiera ju
255	Jakob: Mm
256	VÅRD: den öppna ju, det kan det göra ju
257	Jakob: Mm
258	VÅRD: Men de kanske... man är inte så låst...
259	Jonas: Näe... tror du att sådana modifikation, eller uppdateringar kan ha med sig oönskade effekter?
260	VÅRD: Det kan det ju ha förstås, det kan det ju ha

261	Jonas: På vilket sätt då?
262	VÅRD: Ja man ändrar en viss funktion så påverkar det en annan så blir det fel, det är en kedjereaktion där ju... fast...
263	Jakob: Mm, låt säga ifall ni hade haft öppen programvara i er verksamhet idag, och så ville ni ändra någonting... så har ni säkert krav på att det ska testas utförligt innan det implementeras, ehh... Finns det rimliga möjligheter att göra en sådan på öppen programvara... Att utföra själva testningen
264	VÅRD: Ja det tycker jag att det finns ju
265	Jakob: Mm
266	VÅRD: Man får ju göra testprotokoll där också så att säga... Sen göra genomföra tester, både sådana här... ehh... programtest, alltså skrivbordstest, först ju, och sen gör man ju användartest, så att säga. Och acceptanstest, så det är ju två steg ju.
267	Jonas: Mm.
268	VÅRD: Och det, det måste man väl... alltså... skillnaden där om man har öppen programvara det är ju det att man måste nog lägga ned lite mer tid på själva testandet ju, testfasen blir ju mer omfattande än standard programvara.
269	Jonas: Mm
270	Jakob: Vad skulle det kunna bero på?
271	VÅRD: Ja det är väl lite det som jag sa... det här... om vi jämför med det här journal system vi har idag, så fort vi gör en minsta ändring, eftersom vi har byggt helt och hållet själva så måste vi vara väldigt noga med test när det blir ny versions byte och så då... på, programvaran så att säga, så att allting hänger med. Men sjukhuset som kör standard journalsystem, dom får det, oftast mer testat av leverantören som har gjort den grundläggande testen först. Man måste kanske ändå göra någon egen test där men man behöver inte göra den så omfattande.
272	Jakob: Mm
273	Jonas: Tror du att den öppna programvaran är... ehh... lättare att anpassa efter er verksamhet
274	VÅRD: Ja de tror jag, de tror jag
275	Jonas: jämfört med slutet
276	Jakob: Låt säga då att... ett tanke exempel här, ifall ni hade bestämt er för att använda öppen programvara, så ska det installeras, över hela verksamheten.
277	VÅRD: Ja just det
278	Jakob: något program, det spelar ingen roll, ja... Själva installationsmomentet där... Hur omfattande skulle det bli tror du, skulle de?
279	VÅRD: Ja de, där kör ju vi... vi kör ju, alltså vi har ju centrala databaser för hela [företagsnamn] så att säga... så att, så mycket mer omfattande blir det kanske inte för öppen programvara
280	Jakob: Mm
281	VÅRD: Men... Det hade varit svårare om man hade haft en mer sådan decentraliserad organisation med lokala IT-applikationer ute... då hade det nog varit svårare kanske med öppenprogramvara men som vi har det... centraliserat
282	Jakob: Mm
283	VÅRD: Alltså vi kör över hela Skåne så har vi ju ändå centrala system och databaser
284	Jakob: Mm
285	VÅRD: Vi har ju ett journalsystem va... så att säga
286	Jakob: Mm, och sen distribuerar ni ut det över...
287	VÅRD: Näe, alltså det är ju då alla arbetsstationer och klienter, de kopplar ju upp sig mot en central server
288	Jakob: Mm ja okej

289	VÅRD: så vi kör ju inga lokala system, däremot i primärvården som... de är ju också spridda över hela Skåne... vårdcentralen
290	Jakob: Mm
291	VÅRD: Men dom har ju valt citrix lösningen så dom har... har bara tunna klienter ute
292	Jonas: Mm
293	VÅRD: Men vi kör ju med fulla... fulla klienter... feta klienter ute så att säga, men vi kör ändå med centrala data... centrala... mot centrala servrar.
294	Jakob: Mm okej
295	Jonas: Om vi skulle göra ett byte, från sluten till öppen programvara
296	VÅRD: Ja
297	Jonas: Hur omfattande tror du att det skulle bli i sådant fall, skulle det kräva väldigt mycket resurser eller skulle det vara lätt?
298	VÅRD: Näe jag tror att blir rätt omfattande ändå, det tror jag... Trots allt där med... Om man skulle göra det bytet
299	Jonas: Vad tror du det har grund i?
300	VÅRD: Ja man måste nog bygga upp strukturen på lite annorlunda sätt... och ehh... Ja det blir ju byte... Man måste anpassa alla applikationer och sådant också ju
301	Jonas, Jakob: Mm
302	VÅRD: Så jag tror det blir rätt omfattande på det sättet.
303	Jonas: Mm
304	Jakob: Ehh...
305	VÅRD: Säkerheten måste man ju se över, alltså driftsäkerhet och allt sådant som
306	Jonas, Jakob: Mm
307	VÅRD: de måste... det är en hel del att tänka på
308	Jonas: Ja.
309	Jakob: Lite sammanfattningsvis... vi har gjort även två intervjuer innan denna då...
310	VÅRD: Ja
311	Jakob: Vi ska jämföra lite olika branscher och så
312	VÅRD: Vilka andra branscher har ni varit på?
313	Jakob: Vi har varit i... på en bank och på en... ett systemutvecklingsföretag
314	VÅRD: Jaha...
315	Jakob: Ehh så vi försöker fånga in så olika spridda skurar så att säga
316	VÅRD: Mm
317	Jakob: Ehh och det råder ju lite olika uppfattning beroende på var man kollar då...
318	VÅRD: Ja just det
319	Jakob: Men, det verkar ändå vara... på något sätt som att... lite märkligt och kanske fråga på detta jag menar nu... ehh det verkar ändå finnas att man tycker att öppen programvara är okej, rent kvalitetsmässigt men så finns det alltid ett men.

320	VÅRD: Ja precis.
321	Jakob: Någon grej...
322	VÅRD: Ja, det är få som vågar. Alltså där finns, det är ett litet men som man hänger upp sig på och så vågar man inte ta steget riktigt
323	Jakob: Och det här men, det är väl i så fall det som förändras beroende på var man tittar lite grann... som banken då, kanske var säkerhet just stort för dom
324	VÅRD: De har ju stor säkerhet
325	Jakob: Ja precis och så
326	VÅRD: Vi har ju då... i sjukvården är det ju framförallt sekretessen som är viktig va.
327	Jakob: Mm
328	VÅRD: Så man kan bygga upp sekretessregler så att... ehh, det är ju viktigt så att t.ex. den som bara är behörig att sköta om patienten ska bara ha behörighet till journalen, då... ingen annan
329	Jonas: Mm
330	VÅRD: Vi har ju det här kändissyndromet ju, om det kommer in en kändis på ett sjukhus så rusar ju alla dit och ska kolla i journalen.
331	Jakob: Mm
332	VÅRD: Det får ju absolut inte ske, så då är det viktigt att man har sekre... hög sekretess skydd i systemen också ju
333	Jakob: Mm
334	Jonas: Kan du i så fall se att öppen programvara skulle misslyckas med det...
335	VÅRD: Näe, jag kan inte säga misslyckas men att det kanske... man får ju bygga upp dom sekretessreglerna själva kanske
336	Jonas: Mm
337	VÅRD: Men det är viktigt i alla fall, och vi här på [företagsnamn]... Vi har ju eget sekretessområde så... Hela [företagsnamn] är ett sekretessområde, sjukvården i sig... övriga är ett sekretessområde. Så när vi skickar en... Om det är en läkare som vill se en journal, som är här, som vill se en journal från en patient som är på sjukhuset så får han inte det ju, det är ett annat sekretessområde
338	Jonas: Mm
339	VÅRD: Så att då måste han ha ett medgivande från patienten att han får titta på hans journal från sjukvården
340	Jakob: Mm
341	VÅRD: Så där är då sekretessregler som, de måste man ju kunna ha i öppen programvara också ju...
342	Jonas: Mm
343	VÅRD: Och det misstänker jag att man får själv bygga upp en hel del då.
344	Jakob: Det är lite intressant ehh... att vara ute och prata med olika personer såhär för man får ändå höra att funderingarna har funnits och finns
345	VÅRD: Ja
346	Jakob: och så vidare... Ehh det är möjligt att det har förändrats över tiden men nu är detta bara en engångsstudie, där vi går ut i loppet om en vecka här och frågar lite olika och så...
347	VÅRD: Ja
348	Jakob: Men det är ändå spännande att höra diskussionen

349	VÅRD: Vi har ju haft mycket... alltså vi har ju ett IT-råd... Jag sa att jag var IT-Chef här, på en förvaltning, där är 12 andra. Vi har ju ett IT-råd med 12, 13 och just det här med Open-Office har ju varit väldig diskussion om man skulle satsa på det.
350	Jakob: Mm
351	VÅRD: istället för MS Office ju
352	Jakob: Mm
353	VÅRD: och det störtade ju på att det inte gick att integrera till vissa journalsystem så att...
354	Jakob: Mm
355	VÅRD: Som Siemens som levererar vårt största journalsystem, de ser ju till att det anpassar sitt journalsystem till Microsoft ju, för Microsoft är en sådan stor spelare så att...
356	Jakob: Mm
357	VÅRD: Där är nog den stora, där är nog den stora haken för vår del. Våra leveranser, våra leverantörer och övriga sektioner anpassar sig till den här stora bakgrunden som Microsoft och kanske till viss mån Oracle och sådana stora...
358	Jonas, Jakob: Mm
359	VÅRD: Med det får man ju själv lägga upp, om man har öppna system så att säga, då är det ingen som hjälper en på det sättet.
360	Jakob: Mm
361	VÅRD: Så där är den stora haken på det, men då... Det kostar ju mycket pengar att välja slutna system ju, det är stora licens... ja alltså licens kostnaden är ju betungande
362	Jonas, Jakob: Mm
363	VÅRD: Vi har ju licenskostnader på flera hundra miljoner om året här i regionen
364	Jakob: Mm, är det så att ni verkligen upplever de som betungande.
365	VÅRD: Ja dem licenskostnaderna upplever man som mycket betungande
366	Jakob: Mm, för...
367	VÅRD: och de blir ju högre och högre

8.4.4 Skola

1	Jakob: Vi kan börja med att bara kort förklara varför vi är här...
2	S: Ja
3	Jakob: Det är så att vi håller på att skriva en C-uppsats på... vi läser båda två kandidatterminen på systemvetenskapliga programmet på Lunds Universitet.
4	S: Mm
5	Jakob: Och det är så att vi har fått upp ögonen lite för det här med open source, alltså öppen programvara.
6	S: Ja...
7	Jakob: Och vi har stött på en hel del information som pekar på att det finns många besparingar att göra, med att gå över till öppen programvara.
8	S: Mm
9	Jakob: Men trots detta så är det väldigt få som faktiskt använder de ute i företag och organisationer.
10	S: Mm
11	Jakob: Och vi luskar lite i vad detta kan bero på.
12	S: Mm
13	Jakob: Då har vi en teori om att det skulle kunna bero på att uppfattningar om mjukvarukvalitet i öppen programvara... att den skulle uppfattas som bristfällig... av vissa personer på lite olika sätt och därför så kommer vi kolla närmare på olika aspekter av mjukvarukvalitet.
14	Jo/S: Mm
15	Jakob: Och se ifall det är nåt speciellt som upplevs brister i mjukvarukvalitet...
16	S: Jag tror personligen att det har med mänskliga faktorn att göra...
17	Jakob: Mm
18	S: För vem är det som bestämmer vilken mjukvara som skall köras på företag och...? Jo det är IT-cheferna. Vilken ålderskategori pratar vi om? 50 plus. Dom har ju nått den delen av livet där man ogärna ändrar på samma sätt som man gör i er ålder. Därför känns det osäkert, man är inte van att tänka i dom banorna. Besparingar: JA. Jag tror dock inte att besparingarna ligger just då, utan besparingarna ligger i att det är öppet format, som därmed inte skjuter in en i en återvändsgränd. Går man helhjärtat in i till exempel Microsofts lösningar, inte sagt att Microsoft är dåliga lösningar men då är du liksom fast där. Du investerar in dig i en slutet system och kan inte ta dig därifrån. Jobbar du med öppna standarder så kan du närsomhelst i princip byta leverantör eller inriktning därför att det är en öppen lösning och dom som stödjer den kan då också ta den vidare.
19	Jonas: Mm
20	S: Det gör det också mycket lättare att byta efterhand, så egentligen är det väldigt konstigt att kommuner använder så lite open source eftersom vi har ett... Alltså där finns ju en lag om offentlig upphandling, var 5:e år så ska det upphandlas nya grejer. I den... I såna avtal får man inte räkna in överflyttningskostnader eller ny-kostnader på grund av att man byter system.
21	Jonas: OK
22	S: Och det borde egentligen göra det ännu mer intressant för kommuner och stat att välja open source för det är mycket enklare att flytta, en open source får jag för mig. Därför det... därför dom skall stödja vissa standarder. Och stödjer man samma standarder så är det väldigt enkelt att få det flyttat.
23	Jonas: Mm
24	S: Alltså det är min personliga... Ja tror inte det har så mycket att göra just med själva själva mjukvaran utan det är just det att man är inte van vid den biten... eh... med det nya tänket, open source.

25	Jakob: Mm det skulle kunna va en... givetvis en anledning så att säga. Nu har vi begränsat oss just på grund av omfattningen på arbetet.
26	S: Javisst, det kan ju bli...
27	Jakob: Ja precis, det finns många aspekter så börjar med att titta på det här och så låter vi det vara öppet för andra att titta på andra aspekter.
28	Jakob: Då har vi ett antal frågor, 20 lite kortare frågor som handlar om helt enkelt dina uppfattningar... Vi vill bara försäkra oss om att du har en del i eller i alla fall en insikt i beslutsfattandet på just den här verksamheten, vid val av programvara.
29	S: Jag har haft och kommer att ha...
30	Jakob: Mm?
31	S: Därför att... eh... Kommun... innan dess att... innan [Ohörbart] 2006, så var det vilda västern när det gäller Lunds Skolor. Varje skola skötte sig själv, dom bestämde över sig själv, då va det jag som ansvarade för visst underhåll, licenser, programvaror, allting när det gäller skolans servrar, klienter... ja...
32	Jonas: På [Företagsnamn] Skolan då?
33	S: Ja, så jag hade liksom hela tiden biten.
34	Jonas: Hur många elever e det då?
35	S: Idag har vi 480, om vi säger cirka 480, om vi inte ska räkna exakt, jag tror 478 senaste räkningen men det kan också tillkommit någon sen dess. Och det kan ha fallit ifrån nån, det är alltid lite så fram och tillbaka. Men vi pratar om 480 elever, från förskolan till åk 9 då.
36	Jakob: Detta berör ju alltså även då anställda... eh...?
37	S: Mm, utöver det så har vi då alltså 70 pedagoger och sen så ytterligare städpersonal och så alltså utöver... som det berör.
38	Jakob: Mm.
39	S: Och egentligen så är jag också ansvarig för dom förskolor, fem förskolor, som finns i området med sina barn, idag. Sen så från 2006 och framåt som samlar man det här i... man tog in teknikerna centralt och man hade en idé om att man skulle bygga upp nånting som heter Plattform 2005. Som skulle då ge stordriftsfördelar, och i viss mån så har man lyckats med detta, licenshantering är betydligt enklare idag. Eh... eftersom man skriver storavtal så behöver man inte ha licensavtal för varje utan man räknar bara antalet maskiner. Det är den typen av lösningar... det är det Microsoft pratar om Enterprise Solutions.
40	Jonas: Mm
41	S: Det gör ju att IT-chefen och dom ansvariga för just licenser då kan sova lite bättre på nätterna. [Här väljer vi att inte transkribera en mening då det anses vara känslig information, inte heller berör den uppsatsens ämne].
42	S: Problemet med den här lösningen är ju att alla burkar ska se likadana ut, och det gör att... alla verksamheter i en kommun är inte lika. En verksamhet i en förskola kontra en fyra, nia, gymnasieklass eller en förvaltningsmaskin är totalt dimentralt olika.
43	Jonas: Mm
44	S: Och beroende på vilken inriktning och vad man jobbar med så skiftar det så väldigt mycket. Därför så tror jag ju inte på just alltså... en lösning.
45	Jonas: Nää.
46	S: Och det fram... den främsta orsaken är ju att vi har i uppdrag att utbilda elever som kommer att arbeta i yrken som inte finns, i branschen som inte ens är påkomna. En enkel sak är bara att prata mobiltelefoni, vi har Sony Ericsson anställer 200-300 personer i veckan, eh i månaden, just nu. Dom har väl mellan 2500, nånstans mellan 2500-3000, jag tror inte dom vet ens själva riktigt hur många anställda dom har just nu där uppe. För 5-6 år sen så fanns det knappt. Alltså det liksom... det går ju så fort.
47	Jakob: Mm
48	S: Och därför så måste vi ju lära eleverna att lösa saker och ting på flera olika sätt.

49	Jonas: Mm
50	S: Jag tror inte på att man har en lösning. Idag vet vi om att det finns en stor mjukvarujätte som dominerar marknaden totalt, men den jätten har redan gått ut och deklarerat att dom har misslyckats. Att dom nu kommer att satsa på virtualisering fullständigt, i framtiden. Och det säger ju bara det att vilken maskin vi har i botten är helt ointressant. Utan vi kommer hämta resursen, antagligen på webben, därför att vi har den resursen idag. Och då kommer... det blir ju en helt annan lösning och det ni pratar om open source ger ju ännu mer möjligheter att kunna påverka user interfacet så att det ser annorlunda ut. Är jag då van vid att det ser annorlunda ut, så påverkar det inte mig ett smack. För jag går också in med öppna ögon, för jag vet vad jag vill göra, och jag vet att det går att göra. Men om knappen sitter där, alltså nere till vänster eller uppe till höger eller vad det nu är, så är det helt oväsentligt.
51	Jonas: Mm, har du någon gång övervägt öppen programvara som ett alternativ?
52	S: Oh ja, många gånger. Jag hävdar att det är den enda vägen vi kan gå, för vi kan inte... vi har ett offentligt uppdrag att serva medborgarna med information och då kan vi inte leverera den informationen i ett slutet format. För man skall kunna öppna den med vilken maskin som helst, med vilken programvara som helst. Idag så finns det inte så många programvaror som... jag såg att Adobe hade fått sitt pdf-format nu godkänt...
53	Jonas: Mm, ISO-certifierat.
54	S: Ja, ISO-certifierat, det är jätteskoj. Det är jättebra att man inte godkände windows-lösningen med en gång. Dom kämpar fortfarande och kommer nog inte ge sig på ett tag. Det är självklart att man ska kunna öppna det på vilket sätt som... Så jag är mycket för open source även om jag inte använt sådär jättemycket så använder jag... jag använder mycket freeware och shareware... nu jobbar ju jag på Mac-plattformen och det ger ju möjligheter till helt andra grejer. Där har man lite liberalare lösningar när det gäller licensfrågor, dela med sig, man har möjligheter att prova saker mycket mer än vad man kan göra i den andra världen, känner jag.
55	Jonas: Mm
56	S: Samtidigt så vet man att den kategorin som kör Mac ju oftast har mer pengar och betalar mer för sig. Och därför så är möjligheterna för tillverkare att öppna... att tillverka program som man enkelt då distribuerar. Med förväntan då att man får tillbaka pengarna utan några licenser och sådär, licensnycklar och dom får ju in sina pengar... är man nöjd med det så köper man det, man vet att pengarna går direkt till dem som har utvecklat det.
57	Jakob: Mm, det är viktigt att poängtera det att dom här frågorna som vi sen kommer ställa... att dom handlar... vi vill gärna att du tänker på... låt säga att vi frågar kanske nånting om kanske funktionalitet och så vidare. Vi vill gärna att du försöker matcha det mot den organisationens behov som finns här så att säga.
58	S: OK
59	Jakob: Så det gäller inget specifikt program, men det gäller dom programmen som ni har... att ni har övervägt eller att ni kör här. Vad gäller öppen programvara?
60	S: Mm
61	Jonas: Sen är vi inte ute efter någon speciell uppfattning utan, vi är inte ute efter att open source ska vara bra eller inte utan...
62	S: Nä det kan ni inte va om ni gör det här i forskningssyfte för ni ska ju gå in med en öppen...
63	Jakob: Precis, precis! Så ja, vi lägger inga värderingar i dina svar över huvudtaget.
64	Jakob: Om vi skulle ta och köra igång?
65	S: Ja.
66	Jakob: Om vi börjar med att diskutera lite funktionalitet, som jag nämnde innan. Om man kollar på dom funktioner som finns i öppen programvara, alltså dom öppen programvara alternativ som skulle lämpa sig för er organisation. Anser du att dom funktionerna motsvarar behoven som finns... eh... som finns här?
67	S: Oh ja... Oh ja...
68	Jakob: Mm?
69	S: Eh, där det finns open source programvara... och det är i så fall... om vi pratar Office-paketet diskussionen, alltså egentligen inte Office-paketet, men ja. Det är ordbehandling, kalkylering, presentation i första hand. Där finns ju flera alternativ som fullständigt fungerar till det fullt ut. Då ska man också vara medveten om att skolorna går in något som heter PIM idag, som är Praktiskt IT och Media kunskap för myndigheten för skolutveckling ligger bakom. Där man

	lär nu upp lärarna att använda den nya tekniken till att använda i klassrummet. Mycket av det bygger från början på Office-paketet därför att dom flesta kommuner hade det. Men den... open... OpenOffice till exempel dyker upp i den här programvaran. Och många andra... alltså man har försökt hitta så mycket programvara som är, citationstecken gratis. För kommunerna... alltså open source lösningar helt enkelt. Så Ja, den frågan är ett definitivt Ja.
70	Jonas: OK. Upplever du att dom funktionerna som finns, att dom uppnår samma resultat? Funktionen kan ju alltid finnas men det är inte säkert att den levererar samma resultat som ett annat alternativ.
71	Jakob: eh, kanske inte samma resultat men det som är...
72	S: För mig, tycker jag inte att det är ett problem. Men jag är relativt datorvan. Vårt problem är det att rätt så stor del av vår lärarkår, om inte alla, i princip är digitala immigranter. Ni ligger ju precis i gränsen att antingen va eller inte va. Nånstans vid 84 har jag för mig man räknar, man drar gränsen nu mellan den nya och den gamla människan.
73	Jakob: Då är det kört för oss...
74	S: Det vill säg e man född med datorerna eller inte. För dom lärarna som inte är datavana så är det helt klart så att det räcker med att flytta en knapp, det vill säga inte en knapp då på tangentbordet utan på skärmen för att det ska bli problem, och strula till sig. Och i sådana fall så kan man få problem med viss open source därför att dominanten inte följer standard. Om vi säger så... Microsoft hittar ju sina egna standarder.
75	Jakob: Mm. Ifall vi säger så... ifall ni skulle ta in en open source-lösning här, har du någon uppfattning om dess förmåga att kunna samverka med annan programvara som ni har?
76	S: Eh... Ja, vi har provat. Vi provade tidigare versioner av OpenOffice och det fungerade utan problem. Det va inga problem att öppna dom dokumenten vi hade gjort eller att skriva egna och spara och öppna upp igen. Jag har sett det ske så det är liksom inga problem.
77	Jakob: Mm.
78	S: Jag träffade dom kommunerna som är inne i PIM förra veckan och där... Örebro kör ju bara open source, där existerar inget annat än open source. Medans myndigheten kör då den store mjukvarujättens lösningar. Mina... eh... ungarna brukar kalla Pyttemjuk numer, det är inte Microsoft, det är pyttemjuk. Eh, och det va inga problem, han hade gjort sina i OpenOffice och det bara startade Pang!
79	Jonas: OK. Har du någon uppfattning om säkerheten i öppen programvara?
80	S: Jag tror inte säkerheten är högre eller lägre i öppen programvara men jag tror att eftersom dom som i största mån sysslar med att just hitta säkerhetshålen är mer positiva till den öppna programvaran, gör att den också hackas mindre. Det är ett av skälen till att Mac inte har haft så stora problem därför att man är mer... man är... man har sympati med den lille och slåss med den store.
81	Jakob: Mm, men skulle du anse att säkerheten är fullgod för den här verksamheten?
82	S: Jada, oh ja. Vi har ju kört Firefox som ju är programvara, eh öppen programvara, och som jag hävdar skall finnas kvar hela tiden. Nu har ju IT-avdelningen bestämt att vi ska ha en grundstruktur och vi ska ha... det är Microsoft Explorer som gäller fullt ut. Vilket å andra sidan har resulterat att vi har problem med den mjukvaran som vi använder som är open source eftersom den är optimerad för Firefox. Vi har haft problem, strul, under hösten där saker inte fungerar i Internet Explorer helt enkelt. Men där... dom som har kört Firefox har inte haft några problem.
83	Jonas: OK.
84	S: Så att... för open source så är det extra viktigt att verkligen följa standarder och att få det att fungera. Samtidigt som att så går det fruktansvärt fort att fixa till saker för att det är så många utvecklare.
85	Jakob: Mm. Ifall vi skulle gå över lite grann på vad vi kallar tillförlitlighet i programvara, eh reliability. Har du någon uppfattning om frekvensen av oväntade avbrott som sker i en sån programvara på grund av buggar?
86	S: Om vi pratar icke-betastadium så tycker jag det är samma sak. Jag tycker inte det är någon större skillnad, jag tycker att... i vissa fall så är man till och med snabbare i buggfixandet i open source. När det gäller beta-versioner så e ju liksom... beta är beta, jag menar det... oavsett vems beta så är det beta av ett skäl, det är inte optimerat ännu. Jag tycker inte det är någon större skillnad, det är samma, om inte bättre.
87	Jonas: Om vi säger att det nu dyker upp ett sånt här oväntat fel i din öppna programvara, har du någon uppfattning om hur programmet kan fortsätta fungera trots det här felet? Så att det inte kraschar bara... att det finns någon felhanteringsförmåga... om du har någon uppfattning kring det?
88	S: Eh, det bygger då på Firefox då som jag har kört mest. Där finns ju en Talk-Back möjlighet som talar in information, på samma sätt som det finns i Microsoft-lösningar att om den går ner ställer den frågan: Vill du skicka

	med en rapport om någonting? Där så kan man då ange vad som har hänt, sen är det ju självklart upp till dig om du vill skicka den informationen vidare eller inte. Men gör du det så går ju den informationen vidare till dom som är intresserade.
89	Jonas: Mm.
90	Jakob: Mm, ifall vi nu skulle säga att det går riktigt illa och programmet eller systemet kraschar helt och hållet, finns det... kräver då öppen programvara för den här verksamheten rimliga resurser för att återställa till det fungerande stadiet? Det kanske inte är så bra att ha Office här i åtanke...
91	S: Nä, alltså problemet är... vi har inga... vi kör inte... det beror lite grann på vilka applikationer vi pratar om. När vi pratar affärskritiska verksamheter så pratar vi om extremt affärskritiska verksamheter, vi pratar om personuppgifter. Vi pratar om betyg, omdömen, det är sånt som inte överhuvudtaget inte får lov att hamna någon annan stans. Man kan tycka att affärsidéer i Volvo och större företag är viktiga men vi pratar om medborgarnas uppgifter, skulle det på något sätt bli en läcka och det kommer ut... Så är det liksom ett problem med väldigt stor dignitet om man säger så. Det gör att dom systemen som sköter det, dom sköts centralt, det är liksom... det har vi inte mycket att bygga på... där finns ingen open source över huvudtaget. Antagligen så skulle man konkurrensätta den fullt ut vilket vi egentligen ska göra så, och kommer snart göra, bland annat då elevregistret måste ju upphandlas igen inom kort så finns det ju nog viss risk... eller förhoppningsvis ska man väl säga att även open source kommer in där. Däremot så ser jag väl ett... har man valt ett system där man möter standarden så att säga så ska det ju inte gå att... om man kör backup... så är det inga större problem att bygga upp det med ett annat system.
92	Jonas: Nä. Om vi tar och tittar lite på användbarheten. Har du någon uppfattning om enkelheten eller svårigheten att förstå den logiska strukturen i ett program? När det gäller open source.
93	S: Det beror lite grann på. Det finns... Det är väldigt svårt att säga... Det finns väldigt bra lösningar och det finns väldigt dåliga lösningar också som jag har sett. Precis som det finns bland dom kommersiella programmen eller vad man ska säga, alltså om man skulle dela upp dom så. Alltså dom slutna systemen och dom öppna systemen. Så finns det ju väldigt stor skillnad där.
94	Jonas: Mm, om man tittar på något program som är typiskt för er verksamhet och dra några slutsatser i så fall?
95	S: För oss alltså... det är mest webbläsare vi pratar om nu. Om man kör Firefox eller Internet Explorer så, det är inte så stor skillnad i funktionaliteten idag. Ett tag kunde man alltid hävda att Firefox var bättre för man kunde köra med fliknavigering men det kan man ju inte göra längre på samma sätt. Eh... nä alltså jag tycker... dom är väldigt lika. Man har hittat en standard som folk har lärt sig, alltså såhär bör en webbläsare se ut. Däremot så måste jag säga att jag är väldigt... jag tycker om den här nya FLOCK. I sitt sätt att tänka, i nya banor, med möjligheten att interagera mer med den nya... med den delen vi kallar för Web 2.0. Där jag kan plocka in MySpace-flikar, facebook, direkt i webbläsaren och ha koll på den, samtidigt som jag kan ha RSS-flöden och annat igång som en del av... Det ska ju vara den sociala webbläsaren som det heter och jag kan hålla med om att den är ju intressant. Den är också... den bygger helt på open source... det är samma motor som vi säger.
96	Jakob: Ska bara tillägga det, jag vet inte om vi sa det innan men... Ifall det skulle vara så att det är någon fråga här som du upplever som... kanske... ja, man har ju inte uppfattningar om allt. Då är det helt OK att du inte har det så att säga.
97	S: Nä...
98	Jakob: Ifall vi skulle gå in på... låt säga att ni väljer en open source lösning här, så finns det kanske vissa kompetenser här och så vidare. Är då öppen program tillräckligt lätt att lära sig, för att det skall gå att använda här?
99	S: Fördelen med öppen program är att den oftast följer standarden. Det är oftast mindre företag som jobbar med den. Nackdelen är att det just är mindre företag som jobbar med den. Det gör också att den är mer... Det finns ju mer känslighet framåt... Men det upp... jag tycker den upplevs inte som nåt nackdel därför att den är öppen och det gör att det är enkelt för mig att byta system så att säga. Har jag information någonstans som följer en viss standard så är det väldigt enkelt att flytta över den till nästa system i så fall, som jag kan se det. Men det där är på gränsfall till vad du sa innan, alltså ifall jag har en uppfattning eller inte...
100	Jonas: Eh, om vi säger att någon här i verksamheten skall utföra en specifik uppgift, du har fått en uppgift som du skall göra. Anser du att det är tillräckligt lätt att göra... utföra uppgiften då med öppen programvara?
101	S: Ja, det är... för skolans värld så är det definitivt ja.
102	Jakob: Eh, en lite mer estetisk fråga här... Har du någon uppfattning om öppen programvaras förmåga att tilltala användaren... alltså rent vad du ser på skärmen så att säga... ifall den är snygg nog?
103	S: Mm, just därför att den öppna programvaran oftast ger möjligheten att anpassa till hur du själv vill ha det, till större del. Teman och den här typen är inte... typiskt en sak som finns i alla nu men som började där därför... det var någon som retade sig på att det såg ut på ett speciellt sätt helt enkelt och ville ha det annorlunda.

104	Jonas: Då kommer vi in lite på...
105	S: Men det är väldigt viktigt det där... estetiken är jätteviktig, mycket viktigare än många tror. Det är inte fel om det ser snyggt ut, tvärt om.
106	Jonas: Då kommer vi in lite på interna kvaliteaspekter kan man säga i mjukvara. Då tänkte vi höra lite kring om du har någon uppfattning om svars och bearbetningstider för exekvering av funktioner i öppen programvara?
107	S: Det har jag inte... alltså det... den lilla exekvering... alltså det kan jag inte svara på tyvärr.
108	Jakob: Då kanske det är svårt att svara även på nästa fråga men jag skall ställa den i alla fall... Ifall du har någon uppfattning om dom resurser som krävs för att köra öppen programvara, alltså i form av hårdvara... eh... vilket i praktiken kanske innebär att ifall ni hade varit tvungna att byta ut någon form utav hårdvara eller annan mjukvara, kanske personal, material i någon annan form och så vidare...
109	S: Alltså själva hårdvaran tror jag inte det är några större problem på egentligen. Det som kan behövas bytas ut är kanske en och annan tekniker, som inte kan systemet som måste antingen läras om eller så. Det skulle va den biten...
110	Jakob: Men... låt säga att ni kanske har... nu vet jag inte hur många ni har... säg att ni har en IT-tekniker, en, två, tre, det vet jag inte... men... skulle det behövs fler personer eller färre eller bara lika många med en annan kompetens?
111	S: Ingen uppfattning, det kan jag inte säg. Jag tror inte det är någon större skillnad egentligen, det handlar om intresset ju. Är man intresserad av att lösa problemet så löser man problemet. Är man inte det så... säger man bara Nej. Det är tyvärr det vanliga med en IT-avdelning.
112	Jakob: Om vi tittar lite på underhållet av en programvara. Vi säger att ni har hittat ett fel, i er programvara. Har du någon uppfattning om hur mycket uppfattning det skulle krävas för att ni skulle identifiera det?
113	S: Nej, jag sitter själv inte med den biten så att det har jag ingen...
114	Jakob: När det gäller att modifiera och uppdatera programvara, anser du att det är tillräckligt... att det går att göra i tillräckligt stor utsträckning?
115	S: För våra behov Ja.
116	Jakob: Mm... Innebär det att det kommer i sådant utförande att det inte behövs uppdateras eller menar du att ni uppdaterar eller uppdaterar det och att det är tillräckligt möjligt?
117	S: Med det menar jag att med dom programvaror som jag har jobbat på som har haft den möjligheten... har oftast haft en inbyggd funktion själv som kollar uppdateringar, med vissa intervaller som är ställda då från början, och så får man en fråga; Vill du uppdatera? Så länge du är administratör på din dator så är det ju inget problem. Det är där problemet ligger just nu med IT-avdelningen för att IT-avdelningen trycker ju ut... paketerar alla program för att få ut dom. Men det betyder ju inte att om jag har installerat en version så betyder det inte att jag automatiskt får nästa version, just nu, därför att dom har inte riktigt koll på den programvaran. Jag tror att hela idén var just med plattformen att när jag säger att jag vill köra ett visst program som jag kan tänka mig... men det har tydligen inte riktigt blivit så, så att det är lite uppror just nu här inom skolans värld.
118	Jonas: Om vi säger att ni tar hem en uppdatering till en programvara som ni använder, har ni någon sorts testning av den då innan ni tar den i bruk?
119	S: Det gör vi inom kommunen Ja.
120	Jonas: OK. Har du i så fall reflekterat över hur möjligheterna ges av öppen programvara att köra tester av den?
121	S: Nej det har jag ingen uppfattning om eftersom jag inte själv gör det. Alltså jag testar ju på min egen maskin så att säga, men det är ju ingen... det är ju ingen labb-lösning... jag testar och ser... fungerar det inte så slänger jag ut det. Då får jag gå tillbaka igen och så får jag tipsa mina användare om att den versionen skall du vänta med.
122	Jakob: Så det kan vara så att en uppdatering kan få oönskade effekter?
123	S: Fast samtidigt så kan man då säga... nu kom jag på en sak här... Det finns ju en hel del just inom öppen programvara så finns det en hel del communities där man jobbar. Vet man att det är en uppdatering så kan man gå in och titta i den innan, är det någon som kört den här innan? Har det blivit någonting, idag? Med den här nya körningen och i så fall vad? Verkar det rimligt så kör jag, verkar det lite... så väntar jag. Det finns ju möjligheter om man är intresserad, just eftersom det är så många som är involverade. Men kanske inte den vanliga vägen som IT-avdelningen väljer det vill säga att dom kan inte ringa ett nummer till supporten, dom får leta på lite olika ställen.
124	Jonas: Nä, men det finns andra vägar och möjligheter?

125	S: Det finns andra vägar Mm... Det är väl det vi kallar för Web 2.0 ? Den icke statiska webben den är... man äntligen börjar dela på alltihop.
126	Jakob: Mm, vi va nog inne på det lite innan med just flyttbarhet av mjukvara. Portability.
127	S: Mm...
128	Jakob: tycker du att öppna programvaru-alternativ är tillräckligt lätta att anpassa efter er verksamhet?
129	S: Än en gång, det som vi använt och det som jag vet finns, nu är det ju väldigt få programvaror vi använder oss av som är ren öppen programvara men jag ser inte något större problem med det.
130	Jonas: Om vi tittar bara på själva installationen av öppen programvara... Har du några uppfattningar där, om den?
131	S: Det är enklare. Därför du laddar ner... du drar ner det direkt från nätet och plockar fram det. Det är liksom inte mer med det. Det är oftast mindre företag och därför har dom inget gigantiska distributionsvägar och den enklaste lösningen är att ha en stream-server någonstans, med ett par speglar...
132	Jakob: Nu kommer en fråga som, tror jag, kan täcka av ett lite större område. Det handlar om ifall ni bestämmer er för att byta ut en så kallad slutna programvara mot en öppen programvara så kan det innebära en viss ansträngning helt enkelt... Har du någon uppfattning om hur stor den ansträngningen skulle bli ifall ni skulle byta?
133	S: Den blir rätt så stor ja... Mest alltså just ur synpunkten ur alltså utbildningssynpunkten. Därför att som jag var inne på innan så har vi rätt så ovana användare inom skolans värld och det gör ju då att ska vi börja med ett nytt system så måste vi ju utbilda dom till att se skillnader mellan dem. Där har du ju, det är det jag menar... det är inte säkert man sparar så mycket pengar kortsiktigt, däremot så tror jag i längden att man gör det eftersom man inte sitter fast i ett system.