



INTERNATIONELLA HANDELSHÖGSKOLAN
HÖGSKOLAN I JÖNKÖPING

Öppen programvara

Undersökning inom högskolor och universitet i Sverige

Filosofie kandidatuppsats inom Informatik

Författare: Anna Rehnström

Melike Tullgren

Handledare: Christina Keller

Framläggningsdatum 2005-06-10

Jönköping 06 2005



JÖNKÖPING INTERNATIONAL BUSINESS SCHOOL
Jönköping University

Open Source Software

A research within universities in Sweden

Bachelor's thesis within Informatics

Author: Anna Rehnström

Melike Tullgren

Tutor: Christina Keller

Jönköping 06 2005

Författarnas tack

Inledningsvis vill vi rikta vårt tack till dem som har gjort det möjligt för oss att genomföra denna uppsats och tacka för att de har delat med sig av sin tid och kunskap;

Christina Keller, doktorand i informatik vid Internationella Handelshögskolan i Jönköping samt vår handledare i denna uppsats

– för mycket bra vägledning, diskussion kring vårt förfarande och stöd under arbetets gång

* * *

Jörgen Lindh, universitetslektor i informatik vid Internationella Handelshögskolan i Jönköping och vår handledares mentor.

– för bra kommentarer och samtal beträffande uppsatsarbetet

* * *

Vi skulle vilja tacka de respondenter som vi har samtalat med under vår uppsats

Ewert Bengtsson, IT-chef vid Uppsala universitet, Joakim Nejdeby IT-chef vid Linköpings universitet, Stefan Henriksson IT-chef vid Högskolan i Jönköping och Håkan Yngvesson IT-samordnare vid Internationella högskolan i Jönköping.

– för vänligt bemötande under uppsatsperioden och öppenhet vid intervjun

* * *

Ett stort tack till dem som har hjälpt oss med korrekturläsning av uppsatsen samt gett oss feedback

Opponentgrupper, IHH

– för den konstruktiva kritik som vi har fått

Kandidatuppsats inom Informatik

Titel: Öppen programvara: En undersökning inom högskolor och universitet i Sverige

Författare: Anna Rehnström & Melike Tullgren

Handledare: Christina Keller

Datum: 2005-06-10

Ämnesord Öppen programvara, Öppen källkod, Fri programvara, Open source software, OSS, IT-investeringar, Diffusion, Högskolor, Universitet

Sammanfattning

Öppen programvara kan enklast beskrivas som programvara där källkoden är tillgänglig, vilket gör det möjligt att modifiera, utveckla, kopiera och distribuera denna om så önskas (Statskontoret, 2004). Enligt Statskontorets hemsida (2004) används öppen programvara främst för att uppnå ett större oberoende av pris- och licensvillkor inom landets statliga institutioner. Eftersom nyttjandet av öppen programvara medför många fördelar anser Statskontoret (2005) att den offentliga förvaltningen ska beakta möjligheterna med denna typ av mjukvara i större utsträckning. I en rapport från Statskontoret (2004) om erfarenheter av öppen programvara inom den offentliga förvaltningen, anges att majoriteten har valt att anskaffa öppen programvara av besparingsskäl.

Vi har i denna uppsats undersökt högskolors och universitets syn på öppen programvara. Detta för att erhålla kunskap om de grundläggande orsakerna bakom olika mjukvaruinvesteringar inom dessa lärosäten. I dagsläget används inte öppen programvara i den utsträckning som är möjlig inom högskolor och universitet i Sverige. Vad beror det på? Vilka faktorer eller kriterier vägs in vid mjukvaruinvesteringar inom dessa organisationer? Hur har utvecklingen av användandet av öppen programvara sett ut på högskolor och universitet i Sverige, historiskt sett? Syftet är att kartlägga orsakerna bakom investeringar i eller avsaknaden av öppen programvara på högskolor och universitet i Sverige.

Då vi haft för avsikt att få en uppfattning om orsaker bakom investeringar i öppen programvara på högskolor och universitet i Sverige, har vi valt att göra kvalitativa intervjuer genom att besöka respektive studieobjekt. Detta då vi varit i behov av information som beskriver sakförhållanden, skeenden och ger uttryck för personliga uppfattningar, för att uppnå vårt syfte och besvara uppsatsens problemställning. När det gäller urvalet av respondenter för undersökningen består populationen av samtliga universitet och högskolor i Sverige, statliga såväl som privata. Vi har valt att begränsa vår undersökning till ett urval av populationen, där samplet består av tre lärosäten som har eller saknar erfarenheter av öppen programvara.

De mest centrala iakttagelserna vi gjort i undersökningen är att samtliga universitet/högskolor som deltagit i undersökningen anger att investeringar i öppen programvara inte beror på besparingsskäl, tvärtom är denna typ av programvara ett dyrare alternativ för läroanstalterna. Något annat som observerats är att flera läroanstal-

ter saknar en tydlig supportfunktion för öppen programvara. Statskontorets (2004) anger dock att öppen programvara främst används för att uppnå ett större oberoende av pris- och licensvillkor. Detta oberoende uppfattar läroanstalterna emellertid som mindre tryggt, då allt ansvar förs över på användaren själv, vad gäller uppdateringar, modifieringar etc. Detta överrensstämmer inte med det som anges i rapporten från Statskontorets (2004) vad gäller orsaker bakom investeringar i öppen programvara.

Vi anser att universitet och högskolor i Sverige borde vara politiskt oberoende institutioner och ha ett ansvar som gentemot studenter. Detta genom att erbjuda produkter inom utbildningarnas områden som inte valts på grund av en stor aktör, vilkens makt på marknaden kan liknas vid ett monopol. Läroanstalterna har enligt vår åsikt en mycket viktig roll vad gäller den fortsatta utvecklingen och användandet av öppen programvara. Vi anser därför att dessa institutioner borde beakta valmöjligheten, inte endast på grund av ekonomiska skäl utan också för att bidra till en ökad konkurrens på mjukvarumarknaden.

Bachelor's thesis in Informatics

Title: Open Source Software: A research within universities in Sweden

Author: Anna Rehnström & Melike Tullgren

Tutor: Christina Keller

Date: 2005-06-10

Subject terms: Öppen programvara, Öppen källkod, Fri programvara, Open source software, OSS, IT-investeringar, Diffusion, Högskolor, Universitet

Abstract

Open Source Software (OSS) can be defined as software where the source code is accessible by anyone who wants to modify, develop, copy and/or distribute it (Statskontoret, 2004). According to the website of the Swedish government agency for public management (2004) OSS is utilized mainly to become more independent of pricing and licensing conditions within the public institutions in Sweden. As the use of OSS comprises a number of advantages, the agency for public management (2005) believes that the public sector should consider the possibility of using this type of software to a greater extent. In a report from the agency for public management (2004) about the experience of OSS within the public sector, it is stated that the majority have chosen to acquire OSS due to reduction in costs.

In this essay we have examined universities view on OSS in order to obtain knowledge regarding the basic reasons behind different software investments within these institutions. Today OSS is not utilized to its full extent within institutions of higher education in Sweden. Why is that? What factors or criteria are taken into consideration when making software investment decisions and how has the use of this type of software evolved over the years within these institutions? The purpose of this essay is to survey the reasons behind investments in, or the lack of OSS within universities in Sweden.

As our intention is to get an idea of the reasons behind investments in OSS within these institutions, we have chosen to make qualitative interviews by visiting each object for the study. We have chosen this method as we have been in need of information which describes the course of events and expresses individual opinions, in order to achieve the purpose of the essay and answer the questions stated in the presentation of the problem. Regarding the selection of respondents for the research, the population consists of all universities in Sweden, both public and private. We have chosen to limit the research to a selection of the population, where the sample consists of three universities which have or lack experience from OSS.

The most important findings we have made in the research is that all universities that have participated states that investments made in OSS is not due to financial reasons, as they believe that larger investments in this type of software instead would increase the costs. Something else that we have observed is that several universities believe that OSS lacks an apparent support function. However, the agency for public man-

agement (2004) states that OSS mainly is used to become more independent of pricing and licensing conditions. This independency is on the other hand looked upon as less secure, by the universities, as the whole responsibility is resting on the shoulders of the user, concerning updates, modifications etc. This does not apply with what is stated in the reports from the Swedish agency for public management (2004) regarding the reasons behind investments in OSS.

We believe that institutions of higher education should be politically independent and have a responsibility towards the students. This, by offering products within the educational areas that are not chosen due to the power of one multinational company, which bears the resemblance of a monopoly. Universities have according to our opinion a very important roll to play in the future of OSS. Hence we think these institutions should consider the option, not only due to financial reasons but also in order to contribute to an increased competition in the software market.

Innehåll

1	Inledning	1
1.1	Problemdiskussion	2
1.2	Syfte	3
1.3	Avgränsning	3
1.4	Disposition	3
2	Metod	4
2.1	Datainsamling	4
2.1.1	Primärdata	4
2.1.2	Sekundärdata	4
2.2	Empirisk undersökning	5
2.2.1	Intervjuer	5
2.2.2	Konstruktion av frågor	5
2.2.3	Val av respondenter och öppen programvara	6
2.2.4	Redovisning av empiri	7
2.2.5	Uppsatsens trovärdighet	7
3	Kunskapsprojektering	9
3.1	Begreppsdefinitioner	9
3.1.1	Öppen programvara	9
3.1.2	Öppen standard	9
3.1.3	Community Source	10
3.1.4	Proprietär programvara	10
3.1.5	IT-investeringar	10
3.2	Kunskapsintressenter	11
3.3	Kunskapskaraktärisering	11
3.4	Kunskapsstrategi	11
4	Referensram	13
4.1	Öppen programvara	13
4.1.1	Vad är öppen programvara?	14
4.1.2	Historik	15
4.1.3	Bakgrund till UNIX/Linux	17
4.1.4	Licenser för öppen programvara	20
4.1.5	För och nackdelar med öppen programvara	22
4.2	IT-investeringar	23
4.2.1	Vad är IT-investerings och beslutsprocess modeller/metoder?	23
4.2.2	Modeller för IT-investeringar	24
4.2.3	Vad är affärsnytta?	25
4.3	Tidigare studier/utredningar	25
4.3.1	IT-användning och strategier vid Universitet och Högskolor	25
4.3.2	Modell för utvärdering av öppen och proprietär programvara	26
4.3.3	Utredningar om öppen/proprietär programvara i Sverige	26
4.4	Diffusionsteori	27

4.4.1	Vad är diffusion?	27
4.4.2	Innovation.....	27
4.4.3	Kommunikationskanaler.....	29
4.4.4	Tid	29
4.4.5	Socialt system	29
4.4.6	Innovation inom en organisation	30
4.4.7	Ytterligare begrepp beträffande acceptans.....	31
5	Resultat/Analys.....	32
5.1	Öppen programvara.....	32
5.1.1	Definition av öppen programvara.....	32
5.1.2	Erfarenheter av öppen programvara.....	33
5.1.3	Användning av öppen programvara.....	33
5.1.4	Varför används/används inte öppna program?.....	34
5.1.5	För- och nackdelar med öppna program	34
5.1.6	Utveckling av befintliga/egna öppna program.....	36
5.1.7	Licenser för öppen programvara.....	37
5.1.8	Attityder gentemot öppen programvara inom organisationen	38
5.1.9	Förutsättningar för användandet av öppen programvara.....	38
5.2	Utvärdering av mjukvara	39
5.2.1	Systemet	39
5.2.2	Organisationen	40
5.3	IT-investeringar	41
5.3.1	Riktlinjer för mjukvaruinvesteringar	41
5.3.2	Principer vid upphandlingar av mjukvara	42
5.3.3	Investeringsprocessen: Idé→ Beslut →Realisering av beslut.....	42
5.3.4	Uppföljning av IT-investeringar	43
5.4	Diffusion	44
5.4.1	Modell för attitydsindelning av organisation.....	44
5.4.2	Kommunikationskanaler.....	45
5.4.3	Tid	45
5.4.4	Socialt system	46
5.4.5	Modell för acceptans av ny teknologi.....	47
6	Slutsats.....	48
6.1	Utvecklingen för användandet av öppen programvara historiskt sett inom läroanstalterna	48
6.2	Orsaker bakom investeringar i öppen programvara inom dessa läroanstalter	49
6.2.1	Erbjuda öppna programvaror till studenter	49
6.2.2	Öppen programvara skapar oberoende.....	49
6.2.3	Källkoden är öppen	49
6.2.4	Säkerhetsmedveten	50
6.2.5	Nyckelpersonsberoende	50
6.2.6	Teknisk utveckling och samarbete.....	50
6.2.7	Relativa fördelar	50

6.3	Orsaker bakom avsaknaden av öppen programvara inom dessa läroanstalter	50
6.3.1	Kostnader	51
6.3.2	Kompatibilitet.....	51
6.3.3	Kompetens	51
6.3.4	Supportfunktion	52
6.3.5	Attityder	52
6.3.6	Användarvänlighet	52
6.4	Sammanfattande slutsatser	53
7	Avslutande diskussion.....	55
7.1	Metoddiskussion	55
7.2	Egna reflektioner.....	56
7.3	Förslag på vidare studier	57
	Litteraturförteckning.....	58

Figurer

Figure 1	Adopter categorization based on innovativeness (Rogers, 1995) ..	28
----------	--	----

Bilagor

Bilaga 1 – Ordförteckning

Bilaga 2 - Modell för utvärdering av öppen och proprietär programvara

Bilaga 3 – Frågeformulär

Bilaga 4 – Empirisk undersökning

1 Inledning

I följande kapitel kommer bakgrund till uppsatsens valda kunskapsområde samt diskussion kring valt problemområde presenteras. Syftet med uppsatsen beskrivs och dispositionen för arbetet redovisas.

Öppen programvara har den senaste tiden uppmärksammats i både media och näringsliv. Därför anser vi att detta kunskapsområde är högst aktuellt och intressant att undersöka. Vi har valt att grunda denna uppsats på de rapporter som statskontoret presenterat beträffande öppen programvara. Enligt statskontorets hemsida definieras öppen programvara enligt följande;

”Med öppen programvara menas programvara där källkoden är tillgänglig, och programvara kan användas, förändras, förbättras, kopieras och distribueras av den som så önskar. Notera dock att öppen programvara inte behöver vara gratis och normalt inte är licensfri – friheter och rättigheter regleras av licensvillkor som förekommer i olika varianter.” (Statskontoret, 2004)

Där anges även att användningen av öppen programvara främst sker för att uppnå ett större oberoende av pris- och licensvillkor inom landets statliga institutioner. Vid upphandlingar stärks då myndigheternas position vilket resulterar i lägre kostnader för mjukvaruapplikationer, och möjliggör ett frigörande av monetära resurser till organisationernas egentliga uppgiftsområden. Tillförlitligheten och prestandan är oftast högre och bättre hos öppen programvara till skillnad från motsvarande proprietära alternativ (Statskontoret, 2005). Med proprietär programvara avses motsatsen till öppen programvara (se begreppsdefinition i avsnitt 1.4.3). De grundläggande friheter som öppen programvara innebär, gäller inte för proprietär programvara och kan alltså lättast beskrivas med det enklare begreppet ofri programvara (Wikipedia, 2004).

De flesta områden inom den offentliga sektorn har idag erfarenheter av övergång till öppen programvara eller programvara grundad på öppen källkod. Eftersom nyttjandet av öppen programvara medför många fördelar anser Statskontoret att den offentliga förvaltningen ska beakta möjligheterna med denna typ av mjukvara i större utsträckning (Statskontoret, 2005).

I en rapport från Statskontoret (2004) om erfarenheter av öppen programvara inom den offentliga förvaltningen anges att majoriteten har valt att anskaffa öppen programvara av besparingsskäl. Enligt rapporten har inte övergången från proprietär till öppen programvara ansetts vara något större problem. De svårigheter som uppstått vid övergången har mestadels varit av organisatorisk art, som till exempel användares motstånd vid förändringar. I rapporten framgår det även att det finns en avsaknad av ekonomiska kalkyler beträffande uppföljning av olika mjukvaror. Enligt Statskontorets rapport (2004) har ett tydligt behov av att utarbeta metoder och modeller för framtida mjukvaruinvesteringar identifierats (Statskontoret, 2004).

Microsofts produkter är idag dominerande på arbetsplatserna för operativsystem och kontorsprogram. Under 2002 omsatte PC-avtalen ca 3 miljarder kronor, vilket innefattar arbetsplatsutrustning såväl som hem-pc, inom den svenska förvaltningen. För

att leverantörerna skall erhålla rabatterade priser måste datorerna vara försedda med ett förinstallerat operativsystem (Statskontoret, 2003). Detta försvårar alternativa investeringar eftersom det skapar ett beroende av pris och licensvillkor.

För att göra det svårare att övergå till alternativa produkter använder ett flertal leverantör avsiktligt inlåsningseffekter. Möjligheten till att konkurrera med företag med en stark position på marknaden och proprietära lösningar minskar därför. Användare av alternativa produkter påverkas av att konkurrensen inte är så omfattande, med högre priser och sämre kvalitet som följd (Statskontoret, 2003). Idealbilden vore att programvaror väljs efter det behov och den funktionalitet som efterfrågas och inte på basis av inlåsningseffekter (Statskontoret, 2003). Inlåsning medför stora problem vid upphandling, drift samt förvaltning av komplexa IT-baserade system, vilka orsakas av kostnader vid byte av system, anpassning till omgivningens system och egen hantering av tidigare fattade beslut (Statskontoret, 2003).

Trots de fördelar som användningen av öppen programvara medför enligt Statskontorets rapporter, så är vår uppfattning att denna typ av programvara fortfarande inte används i den utsträckning som är möjligt inom högskolor och universitet i Sverige. För att få svar på orsakerna till detta måste vi beakta följande problem.

1.1 Problemdiskussion

Vi vill undersöka högskolors och universitets syn på öppen programvara. Detta för att få kunskap om de grundläggande orsakerna bakom olika mjukvaruinvesteringar inom dessa lärosäten. Synen på olika typer av mjukvara spelar en stor roll vid mjukvaruinvesteringsbeslut och vad högskolor/universitet grundar sina investeringsbeslut på kan också ge en bättre förståelse för avsaknaden eller användningen av öppen programvara i verksamheten.

Vi anser att marknaden idag har nått den mognad som behövs för att fullt ut använda öppen programvara och tycker därför att detta är ett högst aktuellt ämne att undersöka. Utvecklingen av öppen programvara sker med hög hastighet och är konstant. Vi anser att det tidigare funnits svårigheter med implementering, användning och förvaltning av denna typ av mjukvara, vilket successivt försvinner i och med teknikens fortsatta utveckling. Detta leder i sin tur till att öppen programvara kan komma att konkurrera ut proprietära alternativ.

I dagsläget används inte öppen programvara i den utsträckning som är möjlig inom högskolor och universitet i Sverige. Vad beror det på? Vilka faktorer eller kriterier vägs in vid mjukvaruinvesteringsbeslut inom dessa organisationer? Hur har utvecklingen av användandet av öppen programvara sett ut på högskolor och universitet i Sverige, historiskt sett?

Dessa frågor/problem ovan är de vi avser att undersöka i denna uppsats. Studien kan sedan komma att vara stöd för främst högskolor/universitet, men också andra organisationer, som underlag vid utvärdering av öppen programvara, men även för att vidga deras vyer vad gäller mjukvaruinvesteringar i allmänhet.

1.2 Syfte

Syftet är att kartlägga orsakerna bakom investeringar i eller avsaknaden av öppen programvara på högskolor och universitet i Sverige.

1.3 Avgränsning

Öppen programvara som kunskapsområde är mycket omfattande och vi har därför bestämt oss för att avgränsa uppsatsen till att göra undersökningen om öppen programvara på universitet och högskolor i Sverige. Till en början var tanken att intervjua sex högskolor/universitet, av dessa var tre statliga och tre privata. Dessvärre var det endast respondenterna på Högskolan i Jönköping, Uppsala universitet och Linköpings universitet som hade tid att ställa upp för en intervju. Detta urval har också gjorts då vi anser att studieobjekten geografiskt sett befinner sig inom rimligt avstånd, ur ett kostnads och tidsperspektiv. Vi har även valt att avgränsa uppsatsen till att titta på följande kategorier av mjukvara: kontorsprogram, serveroperativ och databaser samt portaler och webbapplikationer. Detta val har gjorts då kategorierna innefattar de mest kända och använda programvarorna av öppen karaktär.

1.4 Disposition

- Uppsatsens första kapitel innehåller en introduktion till det valda kunskapsområdet, beskrivning av problemområdet och frågeställning. Syftet med uppsatsen och vilken avgränsning vi gjort ingår också i kapitlet.
- I uppsatsens andra kapitel diskuteras gjorda metodval utifrån olika vetenskapsteorier.
- I uppsatsens tredje kapitel presenteras begrepp som vi anser vara centrala i uppsatsen. Här analyseras och karaktäriseras även den kunskap som presenteras i uppsatsen.
- I det fjärde kapitlet behandlas den för uppsatsen relevanta teorin. Det teoretiska ramverket består bland annat av en definiering av vad öppen programvara är, historik kring fenomenet, exempel på produkter och licenser samt fördelar och nackdelar. Utöver detta kommer även en modell/metod för utvärdering av öppen programvara presenteras. Teorier rörande IT-investeringar och diffusionen kommer också tas upp.
- I det femte kapitlet presenteras en sammanfattning av den mest relevanta empirin, vilken analyseras med stöd av det teoretiska ramverket.
- I det sjätte kapitlet presenteras de slutsatser som framkommit genom resultatet/analysen.
- I det sjunde kapitlet förs avslutande diskussion vilken innehåller metoddiskussion, reflektioner och förslag ges på vidare studier inom ämnesområdet.

2 Metod

I följande kapitel kommer förfarandet vid insamlingen av information redovisas och olika definitioner och kategorier av data beskrivs och diskuteras. Vi kommer också att presentera tillvägagångssättet för vår empiriska studie där genomförande av intervjuerna, konstruktion av frågor, val av respondenter, redovisning av empirin samt uppsatsens trovärdighet ingår.

2.1 Datainsamling

2.1.1 Primärdata

Primärdata är den data som består av försthandsinformation som forskaren själv tagit fram. Primärdata kan vara observationer, intervjuer, enkätundersökningar etc. Dessa primärdata kan vidare vara av kvalitativt eller kvantitativt slag. Kvalitativ ansats betyder att en mer detaljerad och mångfacetterad bild av undersökningsområdet försöker ges med hjälp av de olika ovanstående metoderna. Kvantitativ ansats går ut på att informationen som insamlas endast ger en övergripande beskrivning och är mätbar (Lundahl & Skärvad, 1999). Primärdata som har redovisats i uppsatsen har samlats in under den empiriska undersökningen, i form av intervjuer av kvalitativt slag.

2.1.2 Sekundärdata

Sekundärdata är den data som består av redan existerande material, såsom böcker, avhandlingar, tidskrifter, och har insamlats vid tidigare utförda empiriska studier (Lundahl & Skärvad, 1999). Utöver dessa sekundärdata har uppsatsen använt sig av tidningar, webbsidor och databaser.

I den initiala informationssökningsprocessen användes Jönköpings högskolebiblioteks söktjänst Julia. Detta för att ta fram ett urval av böcker och uppsatser som kunde vara relevanta för uppsatsämnet. Några av de sökord som användes var Open source software, öppen källkod, öppen programvara, Linux, Microsoft, Windows, Högskolor, Universitet, Closed source software, IT-strategier, IT-investeringar. Den stora mängd av sekundärdata som fanns tillgänglig krävde att en större avgränsning gjordes, och detta genomfördes i samråd med uppsatshandledaren.

Enligt Christensen et al. (2001) finns det ett antal fördelar och nackdelar med sekundärdata. De fördelar som finns är att nyttjandet av sekundärdata är kostnadseffektivt, tidseffektivt samt att tillgången till källor är omfattande. Nackdelarna som Christensen et al. (2001) tar upp är att det ibland saknas användbar och aktuell data.

Dessa för- och nackdelar gäller generellt för insamling av sekundärdata, men alla kan inte härledas till vår uppsats. Det som överensstämmer med våra erfarenheter är att det är tidseffektivt att använda sekundärdata och att det finns ett stort urval, men ibland finns det bitvis inaktuell data och i vissa fall vinklad och ej pålitlig data. Detta gäller speciellt Internetreferenser och därför har vi varit mycket observanta när det

gäller denna sorts informationsinsamling, för att presentera så trovärdiga källor som möjligt.

Även om användandet av sekundärkällor betecknas som tidseffektivt har vi ändå lagt ner avsevärd tid för att insamla sekundärkällor som är relevanta för uppsatsämnet. Detta för att få trovärdiga, nyanserade och intressanta källor som gör det möjligt att ge en objektiv kunskapsgrund för vårt teoretiska ramverk. För att öka tillförlitligheten i vårt teoretiska ramverk har vi använt oss av olika teorier på samma uppsättning empiri, detta kallas enligt Jensen (1995) teoretisk triangulering.

2.2 Empirisk undersökning

2.2.1 Intervjuer

Vi har för avsikt att få en uppfattning om orsaker bakom investeringar i öppen programvara på högskolor och universitet i Sverige. Vi har valt att göra kvalitativa intervjuer genom att besöka respektive studieobjekt. Då vi talar om studieobjekt avser vi högskolor och universitet och inte enskilda individer. Skälet till att vi valt att göra kvalitativa intervjuer är att vi behöver mer detaljerad information än vad, till exempel en enkätundersökning, som är av kvantitativ art, skulle ge. Kvalitativa undersökningar medför ett nära och direkt förhållande till det som studeras, möjligheter till att karaktärisera och beskriva det ingående. Till skillnad från en kvantitativ undersökning där intervjuer är mer strukturerade och systematiska, tillåter kvalitativa intervjuer flexibilitet, öppenhet och improvisation (Seigerroth, 2005). För att uppnå vårt syfte och besvara uppsatsens problemställning är vi i behov av information som beskriver sakförhållanden, skeenden och ger uttryck för personliga uppfattningar. Detta är också avsikten med kvalitativa intervjuer enligt Repstad (1999). Intervjuerna kan enligt Lundahl och Skärvad (1999) ha olika grad av standardisering och strukturering, vilket avser intervjuarens respektive respondentens frihet under undersökningen. I vår undersökning kommer vi att använda den semistandardiserade metoden. Det innebär att frågor kommer att ställas utifrån en bestämd mall, men att denna inte nödvändigtvis måste följas till ”punkt och prick”, utan utsvävningar från mallen är tillåtna, i form av omformuleringar och följdfrågor. Vi har valt denna metod eftersom att vi genom de förbestämda frågorna vill ha möjlighet att dra generella slutsatser bland respondenterna men också ha möjlighet att beakta det specifika för respektive studieobjekt.

2.2.2 Konstruktion av frågor

Då vi formulerat intervjufrågor har vi utgått från uppsatsens teoretiska ramverk och låtit det styra de kunskapsområden vi avser att undersöka. Vi har framställt ett frågeformulär som gör det möjligt att uppfylla syftet med uppsatsen. Formuläret har utformats med stöd av Repstads (1999) trattprincip. Denna princip går ut på att frågeformuleringen initialt är på en mer generell nivå men efterhand blir mer och mer detaljerad.

Vi inleder formuläret med frågor kring studieobjektens definition av fenomenet öppen programvara och hur de ser på möjligheterna/svårigheterna med användandet av

det. Vidare kommer frågor kring mjukvaruinvesteringar ställas, där en redogörelse för beslutsprocessen efterfrågas. Avslutningsvis ställs frågor om diffusion, det vill säga spridning av nya tekniska innovationer. Konstruktionen av frågorna i uppsatsen har en logisk ordning som är lätt att följa och koppla till vår teori, vilket skapar en röd tråd genom hela arbetet.

2.2.3 Val av respondenter och öppen programvara

När det gäller urvalet av respondenter för undersökningen består vår population av samtliga universitet och högskolor i Sverige, statliga såväl som privata. Vi har valt att begränsa vår undersökning till ett urval av populationen, där samplet består av tre lärosäten som har eller saknar erfarenheter av öppen programvara. Detta urval består av två statliga respektive en privat läroanstalt/er. De läroanstalter som vi valt att intervjua är Högskolan i Jönköping, Uppsala universitet och Linköpings universitet.

Linköpings Universitet grundades 1975 och var då Sveriges sjätte universitet. Idag har universitetet cirka 26 500 studenter och cirka 3 500 anställda och läroanstaltens omsättning uppskattas till 2,5 miljarder kronor i år (2005). Universitetet erbjuder 100 utbildningsprogram och ungefär 500 fristående kurser. Det finns fyra stycken fakultetshögskolor vilka är; filosofiska fakulteten, hälsouniversitetet, tekniska högskolan och utbildningsvetenskap.

Uppsala universitet grundades 1477 och är nordens äldsta universitet. Universitetet är statligt och är ett av landets största universitet med cirka 40 000 studenter och cirka 6000 anställda. Skolans omsättning uppskattas till 4,4 miljarder kronor i år. På Uppsala universitet finns ett stort utbildningsutbud och drygt 40 utbildningsprogram och mer än 1 800 kurser erbjuds. Utbildning och forskning är uppdelat på nio olika fakulteter, teologiska, juridiska, historisk-filosofiska, språkvetenskapliga, samhällsvetenskapliga, utbildningsvetenskapliga, medicinska, farmaceutiska, teknisk-naturvetenskapliga fakulteterna.

Högskolan i Jönköping grundades 1994 och är en stiftelsehögskola (privat). Skolan kom till under borgerligt styre och var ett "experiment" som delvis finansieras av staten och delvis av privata investerare. Eftersom Högskolan i Jönköping inte är helägd av staten betyder detta att det finns utrymme för mer frihet än på andra lärosäten. Högskolan i Jönköping har cirka 8 000 studenter och cirka 700 anställda. Läroanstaltens omsättning uppskattas till 562 miljoner kronor i år. Högskolan erbjuder drygt trettio program och ungefär 300 fristående kurser inom hälsa, vård och socialt arbete, undervisning, medie- och kommunikationsvetenskap, teknik, naturvetenskap och ingenjörsvetenskap samt ekonomi, juridik och informatik. Högskolan i Jönköping består av fem juridiska enheter, fyra fackhögskolorna och högskoleservice.

Vad gäller val av öppen programvara finns ett flertal olika kategorier. Vi har gjort en avgränsning där vi avser att undersöka öppen programvara indelat efter: kontorsprogram, serveroperativ och databaser samt portaler och webbapplikationer.

2.2.4 Redovisning av empiri

Respektive intervjuobjekts svar har sammanställts och redovisats separat i bilaga 4. Varje intervju har redovisats i en löpande text för att underlätta för läsaren då vi söker kontexter mer än detaljer och enskilda händelser. Vi har valt att redovisa intervjuerna som bilagor då vi anser att den mest centrala informationen har presenterats och analyserats i resultat/analys kapitlet. I detta kapitel representeras respektive läroanstalts ställningstagande och åsikter beträffande fenomenet öppen programvara.

2.2.5 Uppsatsens trovärdighet

Eftersom denna uppsats använder sig av en kvalitativ datainsamlingsmetod är det svårt att applicera begrepp som validitet och reliabilitet då dessa i större utsträckning förknippas med kvantitativa datainsamlingsmetoder. Enligt boken "Grundbok i forskningsmetodik" av Patel och Tebelius (1987), anges andra begrepp som den kvalitativa datainsamlingsmetoden kan identifieras med. Motsvarigheten till validitet i kvantitativa undersökningar är rimlighet i kvalitativa dylika. På samma sätt används trovärdighet i kvalitativa undersökningar istället för reliabilitet.

Intervjuerna på varje lärosäte har genomförts med hjälp av personer som är anknutna till de problem som vi vill diskutera, och vi har genom semistandardiserade frågor genomfört intervjun. För att kontrollera rimligheten i intervjusvar ska forskaren jämföra svar på frågor om fakta med officiella uppgifter. Forskaren skall också ställa flera olika frågor om samma sak under intervjun, för att i efterhand jämföra svaren (Patel & Tebelius, 1987). För att undersöka om svaren i intervjun är rimliga används, som tidigare nämnts, den semistandardiserade intervjuguiden. Detta eftersom den tillåter att ställa följdfrågor utifrån de förutbestämda frågorna, för att därmed kontrollera rimligheten. För att ytterligare kontrollera rimligheten i intervjun kan redovisningen av intervjun också granskas av intervjuobjekten så får möjlighet att avgöra om intervjun är riktigt återgiven. Detta har i denna uppsats skett, och alla intervjuobjekten har tagit del av sin egen intervju för att få chans att kommentera och ändra på den.

Trovärdighet mäts genom flera olika åtgärder, för det första måste forskaren förhålla sig objektivt vid intervjutillfället så att informationen som framkommer är oberoende av forskaren. Forskaren måste, för det andra, vara medveten om så kallade bedömarfel som kan uppstå vid en kvalitativ intervju. Det finns flera olika bedömarfel men det mest påtagliga är haloeffekten vilket innebär att det första intryck forskaren får kommer att påverka dennes bedömningar under hela intervjun. För att undvika bedömarfel så ska bandspelare användas vid intervjutillfället, och möjligtvis en oberoende bedömare (Patel & Tebelius, 1987). För att undvika haloeffekten har vi varit två personer som ställt frågor vid intervjutillfället, för att efteråt diskutera uppfattningar om intervjuobjektet. Detta för att undvika att en persons intryck färgar intervjun. Därefter ska också frågorna vara utformade så att de inte kan uppfattas felaktigt. Tidsmarginaler måste också kontrolleras innan intervjun genomförs (Patel & Tebelius, 1987).

För att kontrollera trovärdigheten i en kvalitativ intervju ska hela den totala situationen registreras, därför är det viktigt att notera det som inte sägs men som istället kan

Metod

avläsas under intervjun, det vill säga attityder och reaktioner. Också ur en trovärdighetssynpunkt är det viktigt att låta intervjuobjekten granska intervjun för att eventuellt upptäcka fel eller liknande.

Om populationen är omfattande måste forskaren välja ett representativt stickprov som är slumpmässigt uttaget. Vad gäller val av intervjuobjekt skall detta baseras på bestämda kriterier, det vill säga till exempel position i gruppen (Patel & Tebelius, 1987). I denna uppsats skall IT-chef och IT-samordnare intervjuas för att begränsa populationen.

3 Kunskapsprojektering

I följande kapitel kommer vi att redogöra för de olika typer av kunskap som skall utvecklas i denna uppsats, en strategi för kunskaps utveckling läggs också fram.

Med kunskapsprojektering menas att planera och utveckla kunskap systematiskt. Denna ansats har utvecklats av Göran Goldkuhl (1998) för att strukturera upp arbetsgången vid utvecklandet av kunskap. Vi har dock endast tagit med de delar som är relevanta för uppsatsen.

3.1 Begreppsdefinitioner

För att klargöra betydelsen av olika centrala begrepp i uppsatsen har vi stipulerat dessa, vilket betyder att en beskrivning av respektive fenomen gjorts utifrån ett visst perspektiv.

3.1.1 Öppen programvara

Vi har i denna uppsats valt att använda begreppet ”Öppen Programvara” som en samlad benämning på de amerikanska begreppen ”Open Source Software” och ”Free Software”. Andra svenska översättningar är fri programvara och öppen källkod (Steria, 2003).

Då vi talar om öppen programvara i denna uppsats avser vi programvara där källkoden är tillgänglig. Denna typ av programvara kan användas, modifieras, vidareutvecklas, kopieras och distribueras av den som så önskar. Det är dock viktigt att notera att öppen programvara inte nödvändigtvis behöver vara gratis och i de flesta fall inte är licensfri. Det är licensvillkoren som reglerar friheter och rättigheter beträffande nyttjandet av öppen programvara (Statskontoret, 2004). I referensramen kommer vi att närmare reda ut de ovan nämnda begreppen för att skapa en tydligare bild av vad vi avser då vi resonerar kring öppen programvara.

3.1.2 Öppen standard

”Det finns ingen klar definition av vad en öppen standard är. Allmänt brukar det dock anses att en standard är öppen om utvecklingen av standarden har skett och sker med en konsensusmodell som inte är styrd av en enskild intressent och där det i princip är möjligt för alla intressenter att ta del av eller delta i arbetet med att ta fram standarden.” (Statskontoret, 2005)

En öppen standards karaktäristiska drag är att det finns en ansvarig förvaltningsorganisation som underhåller, dokumenterar och daterar standarden. Den ska också finnas tillgänglig för allmänheten, gratis eller till självkostnad (Statskontoret, 2005).

Öppen källkod har inget med öppna standarder att göra per definition, utan bestäms av oberoende standardiseringsorgan. Ett av de centralaste standardiseringsorganen heter W3C eller World Wide Web Consortium, vilket är en organisation som består av en stor mängd företag som gemensamt kommer överens om de tekniker som ska bli standard på webben. Två språk som används för att visa och bearbeta information på

Internet är HTML och XML, vilka är exempel på öppna standarder (Microsoft, 2004).

3.1.3 Community Source

”Community Source” skapar en gemensam besittning av källkod vitt tillgänglig, precis som öppen programvara modellen gör, men det finns två betydelsefulla skillnader: ”Community Source” modellen kräver kompatibilitet bland tidigare utvecklade versioner av programvara och detta upprätthålls genom testning av proprietära modifieringar och vidareutvecklingar inkluderat prestandaförbättringar. Dessa viktiga skillnader och andra detaljer gör ”Community Source” till en mäktig kombination av de bästa proprietära licenserna och modernare än licensieringsmodellerna för öppen programvaruteknologi (Gabriel & William, 2005).

3.1.4 Proprietär programvara

Något som är proprietärt (engelska: proprietary) är något som har en ägare. Proprietär mjukvara är en försvenskning av det engelska proprietary software.

Med proprietär programvara avses motsatsen till öppen programvara. De grundläggande friheter som öppen programvara innebär, som till exempel, tillgång till koden, möjligheter till modifieringar av programvaran och distribuering av kopior av denna, gäller inte för proprietär programvara. Proprietär mjukvara kan alltså lättast beskrivas med det enklare begreppet ofri programvara

Öppen programvara har dock också en ägare: denne har givit användaren vissa friheter (och därmed följande skyldigheter, se GPL, avsnitt 4.1.4) och ser genom sin upphovsrätt till att dessa efterlevs. Exempel på proprietär programvara är Microsoft produkter som t.ex. operativsystemet Windows NT, webbläsaren Internet Explorer, etc. (Wikipedia, 2004).

3.1.5 IT-investeringar

Med termen IT-investeringar avser vi kostnader associerade med inköp av datorer, kommunikation, nätverk och personal som kan förvalta och driva ett management information system (MIS).

Ett MIS består av personal som förvaltar och driver företagets informationsteknologi. Personalen kan bestå av användare som behöver teknologisk utbildning och kan därför också räknas som en IT-investering, personalen med teknisk kompetens vilken hanterar input/output funktioner i systemet och driver de operativa data systemen och deras chefer. Andra komponenter inkluderar applikationsmjukvara som till exempel programmeringskod, assemblykod, C++ etc. och systemmjukvara som till exempel operativsystem, tolkare, kompilator, program för att hantera data etc. Kärnan i denna modell består av drivande interaktiva komponenter som till exempel datorer, hårddiskar och system, kommunikationssystem, nätverkssystem etc. (Schniederjans et al., 2004).

3.2 Kunskapsintressenter

Vi har skrivit denna uppsats ur högskolornas och universitetens perspektiv huvudsakligen för dem, men även för andra som är intresserade av kunskapsområdet av olika skäl. Universitet och högskolor är de största intressenterna, det vill säga tänkbara kunskapsanvändare, men andra intressenter kan även vara Statskontoret, samt andra statliga institutioner och kommersiella företag vilka är intresserade av investeringar i öppen programvara. Statskontoret kan finna denna kunskap intressant då information angående studieobjektens preferenser, synpunkter och värderingar vad gäller öppen programvara presenteras. Detta kan vara särskilt intressant då liknade rapporter/undersökningar tidigare gjorts inom denna organisation. Evaluering av öppen programvara kan vara av intresse för den offentliga förvaltningen för olika ändamål.

3.3 Kunskapskaraktärisering

”Kunskapskaraktärisering är att ange vad det är för typ av kunskap som skall utvecklas.”
Goldkuhl (1998 s.18)

Vår undersökning består delvis av kategoriell kunskap, vilket innebär att dela in världen i olika kategorier av företeelser och karaktärisera fenomen (Goldkuhl, 1998). Detta har vi gjort då vi brett respektive undersökningsobjekt definiera öppen programvara som fenomen samt i vårt teoretiska ramverk där vi stipulerat olika begrepp. Med detta menas att en beskrivning av respektive fenomen gjorts utifrån ett visst perspektiv. Resultatet innehåller också egenskapsbestämmande (deskriptiv) kunskap, där egenskaper beskrivs hos en kategoriserad och studerad företeelse. I denna kunskapskategori kan högskolors och universitets beskrivning av verksamheten, samt användningen av olika typer av programvara ur ett historiskt perspektiv, inkluderas.

Historisk-rekonstruktiv kunskap beskriver ett fall och dess händelseförlopp (Goldkuhl, 1998). I undersökningen skildras händelseförlopp som till exempel införandet av ny programvara samt händelseförloppet vid beslutsprocesser inom organisationen.

Karaktäriserande kunskap tas fram i uppsatsens resultat, vilket både innefattar förklarande och förståelseinriktad kunskap. Förståelsekunskap inriktas på vad något är medan förklaringskunskap talar om varför något är som det är. Lämpliga innebörder av ett fenomen bestäms med förståelseinriktad kunskap (Goldkuhl, 1998). Detta görs i undersökningen genom intervjuobjektens karaktärisering av öppen programvara som fenomen, närmare bestämt anger de ett antal egenskaper. Med förklaringskunskap avses en redogörelse för varför något är på ett visst sätt. Orsaker, grunder, skäl eller förutsättningar för något resulterande förhållande anges (Goldkuhl, 1998). I uppsatsens analys kartläggs orsakerna bakom investeringar i, eller avsaknaden av öppen programvara på högskolor och universitet i Sverige, vilket därför kan anslutas till ovan nämnda kunskapskategorier.

3.4 Kunskapsstrategi

”Strategiformulering innebär att man analyserar och fastställer det principiella tillvägagångssättet för kunskapsutvecklingen.” Goldkuhl (1998, s.23)

Kunskapsprojektering

Uppsatsen har en medveten explorativ ansats vilket innebär att forskningen inriktar sig på att undersöka ett område där den befintliga kunskapen är liten eller näst intill obefintlig. Vi har valt denna ansats då vi vill förbättra vår kunskap inom undersökningsområdet, men ej ämnar testa hypoteser, utan snarare generera dessa. I kombination med den explorativa ansatsen har vi valt att använda en komparativ inriktning på undersökningen eftersom vi avser att jämföra olika företeelser mellan de olika läroanstalterna (Goldkuhl, 1998).

4 Referensram

I följande kapitel kommer vi inledningsvis att beskriva öppen programvara som fenomen och beskriva framväxten av denna typ av mjukvara. Vi kommer även att exemplifiera två av de mest kända produkterna med öppen källkod på marknaden idag. En redogörelse för hur det fungerar med licenser för öppen programvara och vilka som är mest kända presenteras. Fördelar och nackdelar med öppen och proprietär programvara diskuteras för att ge läsaren en bättre utgångspunkt vid val av mjukvara. Vi avser att lägga fram en modell för utvärdering av mjukvara, vilken kommer att ta upp kriterier som är av vikt att beakta vid investeringar i mjukvara. Andra delar som ingår i kapitlet är IT-investerings teorier och modeller, samt teorier om diffusion av tekniska innovationer. Samtliga ovan nämnda delar anser vi vara essentiella för att besvara vår problemställning och uppnå syftet med undersökningen. Vi har utgått ifrån detta teoretiska ramverk när vi utfört vår empiriska studie. Samtliga kursiverade begrepp som presenteras här finns med i ordförteckningen, se bilaga 1.

4.1 Öppen programvara

Mjukvaran, även kallat programvara, är de beståndsdelar i ett datorsystem, vilka är abstrakta, till skillnad från hårdvaran. Mjukvara kan delas upp i två delar; operativsystem och applikationsprogram. Operativsystemet finns i bakgrunden och sköter minneshantering, skrivning till och från diskar, och så vidare och fordras för att över huvud taget kunna använda hårdvaran. Ovanpå operativsystemet körs sedan applikationsprogram som till exempel ord- eller bildbehandlingsprogram, kalkylprogram eller dataspel.

Mjukvara skapas genom att skriva instruktioner konstruerade av ord, symboler och siffror. Dessa kallas källkod och är läsbara för människor. För att kunna köra (execvera) programmet används en kompilator. Detta är ett program som omvandlar källkoden till maskinkod bestående av ettor och nollor. Kompileringen är en process som skapar en binär fil vilken är läsbar för datorer men inte för människor. För att använda datorprogrammet behövs inte källkoden, utan endast den binära filen eftersom att den är det egentliga datorprogrammet (Danielsson, 2002).

För att kunna göra modifieringar i programvaran måste användaren ha tillgång till källkoden, vilken är den kod som skrivs av programmerare. Se följande exempel.

```
#include <stdio.h>

main() {
    printf("Hello world!\n");
}
```

Detta är ett exempel på källkod, vilket är skrivet i programmeringsspråket C

Denna kod kan vara sluten, vilket betyder att den inte är tillgänglig för allmänheten, eller öppen, vilket avser att andra får se den. Därav uttrycket ”öppen källkod” (Wikipedia, 2004).

4.1.1 Vad är öppen programvara?

Vi har i denna uppsats valt att använda begreppet ”öppen programvara” som en samlad benämning på de amerikanska begreppen ”Open source software” (OSS) och ”Free software”. Andra svenska översättningar är öppen källkod och fri programvara (Steria, 2003). I följande text kommer vi att reda ut de ovan nämnda begreppen för att skapa en tydligare bild av vad vi avser då vi resonerar kring öppen programvara.

Enligt grundaren av *FSF (Free Software Foundation)*, Richard M. Stallman, måste följande punkter uppfyllas för att kunna definiera programvara som öppen;

- frihet att nyttja programmet för valfritt ändamål.
- frihet att undersöka hur programmet fungerar och modifiera det för egna ändamål.
- frihet att vidare distribuera kopior av programmet.
- frihet att utveckla programmet och sprida det vidare i förbättrad form.

(Susning, 2004)

Detta betyder att användaren själv kan rätta fel och göra förbättringar i programmet eller anlita ett företag eller någon programmerare som kan göra det.

Ett flertal kommersiella programvaror har inte öppen källkod, vilket skapar ett beroende mellan användare och leverantör, eftersom att vid önskemål om modifieringar och förbättringar är det leverantören som besitter rättigheterna för programvaran och därför den enda som kan göra ändringar i den. Problem kan därför uppstå om leverantören antingen är ointresserad av att göra förbättringarna eller har gått i konkurs. Samtidigt som öppen programvara, på gott och ont, tar bort sambandet mellan en användare och en leverantör, ökar också möjligheterna radikalt att kvalitetsgranska koden (Susning, 2004).

Öppen programvara betyder som tidigare nämnts inte nödvändigtvis att den är gratis och är i regel inte licensfri. De friheter och rättigheter som öppen programvara medför regleras av licensvillkor som förekommer i olika varianter. Dessa kommer att tas upp på en mer detaljerad nivå i avsnitt 3.1.4 Licenser för öppen programvara (Statskontoret, 2004).

4.1.1.1 Öppen källkod vs. Open Source Software

Öppen källkod är den svenska översättningen av Open Source Software och betyder att en programvaras källkod är öppen, det vill säga tillgänglig för användare att läsa eller förändra enligt egna önskemål. Användaren kan på så sätt skraddarsy program-

varan efter eget behov. Vanligtvis erbjuds den ursprungliga upphovsmannen den modifierade versionen, vilken i sin tur kan välja att göra den till en del av den officiella (Wikipedia, 2005).

4.1.1.2 Öppen källkod vs. Fri programvara

Öppen källkod är en annan benämning för det ursprungliga begreppet fri programvara. Skillnader mellan öppen källkod och fri programvara är utan praktisk betydelse för den vanliga användaren. Vid diskussioner kring de två begreppen avses skillnaden vara att öppen källkods rörelsen är mer pragmatisk, medan rörelsen för fri programvara är av en mer politisk art. Den senare använder licenser som till exempel *GPL* (*GNU General Public License*, se avsnitt 4.1.4) vilka deklarerar att fri programvara endast får byggas in i annan fri programvara (Wikipedia, 2005).

Öppen källkods rörelsen väljer istället licenser som inte begränsar användningen av källkoden, såsom *BSD-licensen* (*Berkeley Software Distribution*, se avsnitt 4.1.4.) Denna nyttoinriktade rörelse vill skapa möjligheter för till exempel företag att ta en kollektivt skapad programvara och vidareutveckla denna till en lukrativ produkt utan att själva behöva göra källkoden tillgänglig för andra (Wikipedia, 2005).

Upphovsrättslagen begränsar en användares friheter om upphovsrättsinnehavaren till programmet inte sagt någonting annat. Att släppa en programvara fri under en licens blir därför nödvändigt för att ge användaren mer friheter. Det finns ett stort antal licenser på marknaden idag för öppen programvara och de två som tidigare nämnts, GPL och BSD licenserna, är de mest förekommande (Susning, 2004).

4.1.2 Historik

I inledningsskedet av mjukvarueringen existerade endast öppen programvara. Senare kom proprietär programvara, vilken snabbt kom att dominera mjukvarumarknaden. Detta har idag resulterat i att proprietär programvara, av många människor, anses vara det enda möjliga alternativet. Det är bara sedan en kort tid tillbaka som mjukvaruindustrin har betraktat öppen programvara som ett alternativ igen (Gonzalez-Barahona, 2000).

Då IBM (International Business Machine Corporation) och andra företag sålde de första storskaliga kommersiella datorerna på 60-talet, kom dem med öppen programvara. Öppen, i den bemärkelse att den kunde användas fritt bland användare, då den levererades med öppen källkod som kunde förändras och förbättras. Situationen förändrades dock i slutet av 60-talet efter bolagiseringen av IBM-mjukvara, och redan i mitten av 70-talet var den vanligaste typen av mjukvara proprietär, i det avseende att användare inte hade tillgång till källkoden, vilket medförde att de inte kunde förändra programmen eller var tillåtna att distribuera dem. I slutet av 70-talet och i början av 80-talet, etablerades grunderna till den nuvarande ”fri programvara” rörelsen av två olika grupper. På USA:s östkust lanserades GNU projektet och The Free Software Foundation av en tidigare anställd på *MIT AI Lab* (*Massachusetts Institute of Technology, Artificial Intelligence Lab*), Richard M. Stallman. Målet med GNU projektet var att

bygga ett fritt operativsystem. GPL skapades med avsikten att behålla den mjukvara som producerades, fri, men också för att främja en ökad produktion av fri programvara. Stallman författade även "the GNU Manifesto" där han deklarerade att tillgängligheten till källkod och friheten att distribuera och förändra mjukvara är fundamentala rättigheter (Gonzalez-Barahona, 2000).

Under tiden utvecklades UNIX systemet på USA:s västkust av *The Computer Science Research Group (CSRG)* vid University of California i Berkeley. Ett flertal applikationer byggdes vilka snabbt blev en variant av UNIX kallat BSD UNIX. Detta utvecklingsarbete finansierades främst av kontrakt med *Defense Advanced Research Projects Agency (DARPA)* och ett omfattande nätverk av UNIX "hackers" runt om i världen, vilka hjälpte till med att finna programfel (debugging), förvalta och förbättra systemet. Många gånger distribuerades inte mjukvaran utanför användargruppen vilken förfogade över en *AT&T* UNIX licens. Emellertid, distribuerades det slutligen under BSD licensen i slutet av 80-talet, vilken var en av de första öppna programlicenserna. Dessvärre var alla slutanvändare av BSD UNIX på den tiden tvungna att ha en *AT&T* UNIX licens, eftersom att några delar av kärnan och ett flertal viktiga funktioner vilka var essentiella för ett fullt användbart system fortfarande var proprietära (Gonzalez-Barahona, 2000).

Under 80-talet och början av 90-talet fortsatte utvecklingen av öppen programvara, inledningsvis i ett flertal relativt isolerade grupper. Genom Internet och *USENET* kunde kraftfulla användargrupper skapas. Detta medförde att många av de program som redan utvecklats successivt integrerades, vilket förenade många användargrupperns arbete. Som ett resultat av denna integration, kunde kompletta miljöer byggas ovanpå UNIX med nyttjandet av öppen programvara. I början av 90-talet var hela marknaden för öppen programvara och utvecklingen av mjukvara i allmänhet, redo för förändring. Två händelser av stor betydelse för utvecklingen av öppen programvara utspelades på två olika platser:

CSRG (The Computer Systems Research Group), vid universitetet i Toronto, skapade en enkel version (Net/2) av "BSD UNIX" som senare fick namnet 386BSD, efter en vidareutveckling av en man vid namn Bill Jolitz. Denna version var fri från *AT&T* kod med copyright, och blev snabbt populär bland BSD och UNIX användargrupper. Förutom att den hade en kärna, fanns även många andra funktioner vilka tillsammans utformade ett komplett operativsystem. Arbetet skyddades av BSD licensen vilket gjorde mjukvaruplattformen helt fri. I operativsystemet inkluderades även öppen programvara under andra licenser. I Finland var en student vid namn Linus Torvalds igång med att implementera de första versionerna av Linux kärnan. Inte långt senare samarbetade många människor med att förbättra kärnans användbarhet, genom att lägga till många funktioner för att fullborda GNU/Linux, ett riktigt operativsystem. Linux kärnan och GNU *applikationer* ovanpå kärnan är skyddade av GPL licensen (Gonzalez-Barahona, 2000).

1993 var både GNU/Linux och 386BSD relativt stabila plattformar. Sedan dess har 386BSD utvecklats till en hel familj av BSD baserade operativsystem (NetBSD, FreeBSD, and OpenBSD), medan Linux kärnan utvecklas och används i åtskilliga GNU/Linux distributioner som till exempel; Slackware, Debian, Red Hat, Suse,

Mandrake och många fler. Under 90-talet har många ”öppen källkods projekt” producerat ett stort antal användbara och oftast högkvalitativa mjukvaror. Några av dem är;

- Apache (använt i stor utsträckning som en webbserver)
- Perl (ett tolkat programmeringsspråk eller skriptspråk)
- XFree86 (den mest använda X11 (användargränssnitt) implementation för PC-baserade maskiner)
- GNOME *GNU Network Object Model Environment* och KDE (båda tillhandhåller en konsistent serie av bibliotek och applikationer för att ge den vanliga användaren en mer användarvänlig skrivbordsmiljö)
- *Mozilla* (det öppna programvaruprojektet grundat av Netscape för att bygga en webbläsare), etc.

Av alla dessa projekt så är GNOME och KDE de mest betydelsefulla då de uppmärksammar användbarheten för otekniska människor. Deras resultat är redan synliga och av hög kvalitet, vilka gör det möjligt för alla att dra nytta av öppen programvara. Mjukvaran som produceras av dessa projekt avfärdar den vanliga myten om att öppen programvara huvudsakligen fokuserar på server och utvecklingsorienterade system (Gonzalez-Barahona, 2000).

Slutet av 90-talet är mycket händelserikt och en intressant era vad gäller utvecklingen av öppen programvara. Öppen programvara system baserade på GNU/Linux eller BSD får allmän acceptans och har blivit ett verkligt alternativ till proprietär system, konkurrerande med marknadsledande produkter, som till exempel Windows NT. Inom många branscher är det bästa alternativet redan öppen programvara, ett utomstående exempel är Apache webbserver med över 50% av marknaden. Frisläppandet av webbläsaren Netscape Communicator 1998 gjorde så att öppen programvara uppmärksammades och ledde till en ökad förståelse för fenomenet bland många stora företag. Många företag av olika storlekar utforskar nya ekonomiska modeller för att bli framgångsrika på den konkurrenskraftiga mjukvarumarknaden. Media har också börjat uppmärksamma den tidigare marginella öppen programvara rörelsen, vilken nu inte endast består av individuella ideella organisationer utan också av små och medelstora företag (Gonzalez-Barahona, 2000).

4.1.3 Bakgrund till UNIX/Linux

År 1969 skapades ett operativsystem kallat *UNIX* (Registrerat varunamn från The Open Group). Detta var ett resultat av Ken Thompsons initiativ. Han hade tidigare arbetat vid Bell Labs i New York, (Numera AT&T) och när ett halvfärdigt projekt, programmerat i assemblerkod, kallat Multics hade avslutats ansåg Ken Thomson att denna idé skulle utvecklas ytterligare. Multics projektet ledde fram till att UNIX enbart distribuerades som källkod under tämligen inofficiella former. Under åren 1972-1974 arbetades ursprungsidén om i programmeringsspråket C på University of California/Berkeley, denna version kom att kallas Berkeley Software Distribution eller

BSD. Eftersom Dennis Richie, skapare av språket C, hade bidragit med så mycket kunskap till projektet står han numera som medkreatör till UNIX. UNIX fanns från början i en version men har efter många år på marknaden spridits och utvecklats till en mängd olika versioner. Många personer har bidragit till att utveckla UNIX till vad det är idag tack vare den flexibla utvecklarvänliga miljö som existerar. 1991 var UNIX det mest använda multianvändaroperativsystemet i världen (Willén, 2001).

4.1.3.1 Vad är UNIX?

UNIX är inte bara ett operativsystem utan en hel familj av operativsystem. UNIX som varumärke ägs av The Open group som också tagit fram en standard som ska uppfyllas innan ett program får kallas för UNIX-baserat. Ett annat företag som också arbetar med att utlysa standarder är Common desktop Environment (CDE) men de inriktar sig huvudsakligen på att konstruera fönstersystem. Dessa två organisationer som utvecklar standarder skall ej blandas ihop med företag som implementerar UNIX-system, såsom SUN Microsystems, HP (Hewlett-Packard) med flera (Willén, 2001).

UNIX operativsystem innehåller bland annat ett hierarkiskt filsystem som har en hög modulariseringsgrad och portabilitet. Annat som UNIX rymmer är kompatibla filer med andra UNIX-varianten, input/outputprocesser, möjligheten till att initiera asynkrona processer, det vill säga det krävs ej att sändare och mottagare är på plats samtidigt (Aronssons Datateknik, 2005), och dessutom finns i över hundra undersystem och ett dussintal språk (Willén, 2001).

UNIX är tillverkat som ett "multiuser" system som möjliggör centralt distribuerad datorkraft och så kallade "dumma terminaler", detta betyder att det går att lägga till användarekonton och köra samma program på flera datorer samtidigt. Det går alltså att använda datorkraft mer effektivt med hjälp av ett multiuser system och därmed ökar möjligheten till att göra besparingar (phpBB Group, 2004).

UNIX har, som tidigare nämnts, flera olika varianter som skiljer sig från varandra. Dessa varianter är dock relativt kompatibla och det är enkelt för användare att byta från en variant till en annan. UNIX är inte heller låst till ett visst datormärke och orsaker som dessa har gjort UNIX väldigt populärt. En del operativsystem räknas in i UNIX-familjen utan att följa standarden som "The Open Group" satt. Dessa system är väldigt lika UNIX men är helt oberoende utvecklade jämfört med UNIX-systemen (DVKV, 2003). Under 90-talet har det vuxit fram UNIX versioner specialframtagna för att köras på vanliga persondatorer bland annat *FreeBSD*, *OpenBSD*, *NetBSD*, *Solaris*, *Mac OSX*, *Linux* med flera. Det mest kända operativsystemet är *Linux* (Linux Online, 2005).

4.1.3.2 Vad är Linux?

Linus Torvald studerade vid Helsingfors Universitetet 1991 när han, mer som ett hobbyprojekt, påbörjade utvecklingen av Linux. Efter det har Linux växt med enorma steg tack vare enskilda utvecklarens hjälp och vidareutvecklas idag av tusentals frivilliga programmerare över hela världen, som kommunicerar med varandra via Internet (Linux Online, 2005).

Operativsystemet Linux är programvara med öppen källkod under licensen GPL. Linux är inte baserad på källkod från AT&T:s *System V* som andra UNIX-system, utan är egenutvecklad kod (Toft, 2004).

Linux är i själva verket bara kärnan, resten av operativsystemet bygger på programvara från andra projekt. För att få Linux att fungera krävs inte bara kärnan utan också distributioner. En distribution innehåller kärnan, all grundläggande systemmjukvara, program, och installerare som enkelt guidar igenom installationen (Toft, 2004).

Linux kärnan finansieras bland annat av kommersiella aktörer däribland IBM och Red Hat. Det finns även ett flertal företag vilka säljer sina produkter eller tjänster med anknytning till öppen programvara. Att kommersiella aktörer som IBM och Red Hat lämnar bidrag till Linux-kärnan och att HP har en Linux-strategi, är klara tecken på kommersiella egenintressen (Andersson, 2005).

.

Exempel på distributioner är;

Slackware

Är den äldsta Linuxdistributionen och följer filosofin KISS ("Keep It simple, Stupid"), men detta är inte, som kan antagas, en nybörjardistribution utan syftar till att hålla uppbyggnaden av systemet enkelt. De som föredrar Slackware är mer erfarna användare. Slackware anses vara en mycket stabil och säker distribution och utvecklas fortfarande aktivt (Tempsch's, 1998).

Mandrake

Är en Linuxdistribution som rekommenderas till nybörjare eftersom konfigurationer är automatiska eller görs enkla med översiktliga guider. Mandrake är inriktat på att ta fram skrivbordsupplevelser (Tempsch's, 1998).

Debian GNU/Linux

Är också en Linuxdistribution som är en bra för såväl nybörjaren som den lite mer vane användaren. Debian GNU/Linux använder sig av binärpaket vilket är förkompilerad källkod av en mjukvara, som är avsett för en särskild plattform, i kombination med ett verktyg, kallat apt-get, som automatiskt laddar ned paketen som önskas installeras (och även reder ut beroendekedjor automatiskt). Med detta hjälpmedel förenklas installation, konfiguration, uppdatering och borttagning av mjukvara avsevärt. Det finns många distributioner som baserar sig på Debian GNU/Linux, till exempel KNOPPIS, MEPIS, Ubuntu (Tempsch's, 1998).

Red Hat

Är ett företag som säljer kommersiella Linux distributioner under eget namn, de står bakom distributionen Fedora som är fritt tillgänglig. Red Hat finns ej som gratisversion (Aronssons Datateknik, 2004).

4.1.4 Licenser för öppen programvara

Med det nuvarande juridiska ramverket, definierar licensen, under vilken ett program distribueras, de exakta rättigheterna vilka en användare förfogar över det. Till exempel så har de flesta proprietära programmen en licens vilken tar bort rättigheter för att kopiera, modifiera, låna och hyra ut, användning i flera maskiner, etc. Licenser specificerar oftast att ägaren av programmet är företaget som gett ut det, vilket säljer begränsade rättigheter för användning av det (Gonzalez-Barahona, 2000).

När det gäller öppen programvara så är licenser under vilket ett program distribueras mycket betydelsefulla. Vanligtvis, är bestämmelserna som specificerats i licensen av öppen programvara resultatet av en kompromiss mellan ett flertal uppsatta mål som i viss bemärkelse är motsägelsefulla (Gonzalez-Barahona, 2000).

Bland dem, kan följande vara värda att nämnas:

- Garantin för grundläggande friheter, som återdistribution, modifiering och användning, för användare.
- Försäkra om villkor införda av upphovsmännen (citering av upphovsman/männen i förändrad programvara baserat på dennes arbete)
- Garantera att utveckling av tidigare arbete också förblir öppen programvara.

Upphovsmännen kan välja att skydda deras mjukvara genom olika licenser, enligt nivå med vilken de vill uppfylla dessa mål, och de detaljer de önskar att tillförsäkra. Upphovsmännen kan, om de så önskar, distribuera deras mjukvara med olika typer av licenser genom olika kanaler och priser. Därför väljs licensen av upphovsmannen till programmet mycket omsorgsfullt. Även användare som återdistribuerar eller modifierar programvaran måste med noggrannhet studera licensen (Gonzalez-Barahona, 2000).

Trots att varje författare har möjlighet att använda olika licenser för dennes program teoretiskt sett, är majoriteten av öppen programvara i praktiken distribuerad under en av de vanligaste licenserna som till exempel GPL, LGPL, BSD, etc. En del organisationer har den senaste tiden definierat karaktäristika mjukvarulicenser borde ha för att kvalificera som öppen programvarulicenser. Bland dem är de mest kända "the Debian project" vilket definierar riktlinjer för öppen programvara från Debian och "the open source initiative" vilket definierar öppna programvarulicenser och är baserat på Debian's riktlinjer för fri programvara. Det är mycket påtagligt att pris eller tillgänglighet av öppen programvarukod i sig självt inte är tillräckligt för att karaktärisera en produkt som öppen programvara enligt Debian's riktlinjer. Den viktigaste punkten berör rättigheterna för användargruppen att fritt ha möjlighet att modifiera och återdistribuera koden eller ändringar av denna, enbart med förbehåll om att dessa rättigheter måste ges till alla och ej får återkallas (Gonzalez-Barahona, 2000).

Skillnader mellan olika öppna programvarulicenser bestäms efter den vikt en författare lägger vid följande punkter;

Skydd för öppenhet

En del licenser kräver att alla återdistributörer behåller samma licens vilket leder till att mottagarens rättigheter är desamma även om mjukvaran är erhållen direkt från författaren eller från annan mellanhand (Gonzalez-Barahona, 2000).

Skydd för moraliska rättigheter

I många länder, skyddar lagstiftningen en del moraliska rättigheter som bland annat erkännande av författarskap. En del licenser tillhandahåller även skydd för dessa ämnen, vilket gör dem opåverkbara vid förändringar i lokala lagstiftningar (Gonzalez-Barahona, 2000).

Skydd för en del ägar rättigheter

I vissa fall har den ursprungliga författaren ytterligare rättigheter vilka i viss mening är en typ av proprietära rättigheter (Gonzalez-Barahona, 2000).

Kompatibilitet med proprietära licenser

En del licenser är designade så att de är helt inkompatibla med proprietär programvara. Till exempel kan det vara förbjudet att återdistribuera mjukvara vilken är ett resultat av en blandning av mjukvara under licensen med någon form av proprietär programvara (Gonzalez-Barahona, 2000).

Kompatibilitet med andra öppna programvarulicenser

Vissa öppna programvarulicenser är inte kompatibla med varandra eftersom att kraven för en inte kan uppfyllas om kraven för den andra är tillgodosedda. I detta fall är det vanligtvis omöjligt att blanda olika mjukvara distribuerade under dessa licenser i en och samma programvara (Gonzalez-Barahona, 2000).

Några av de mest erkända öppna programvarulicenserna är följande;

BSD (Berkeley Software Distribution)

BSD licensen täcker inte endast BSD publiceringar utan även andra öppna programvaror. Denna licens ger användaren stor frihet då det nästan inte finns några villkor eller restriktioner för användningen av mjukvaran. Till exempel så kan debitering av klienter ske vid distribution av kod och licensen har även en avsaknad av krav på öppenhet, källkoden måste därför inte inkluderas vid distribution. Sammanfattningsvis kan alltså återdistributörer göra näst intill vad som helst med mjukvaran innefattande användning av denna för proprietära produkter. Författarna vill endast att deras arbete skall bli erkänt. I viss mening kan denna restriktion försäkra en del fri marknadsföring i det avseende att det inte kostar några pengar. Det är viktigt att notera att denna typ av licens inte innefattar någon bestämmelse som garanterar att reviderad mjukvara fortsätter att vara öppen (Gonzalez-Barahona, 2000).

GPL (GNU General Public License)

GPL är en licens som är distribuerats under GNU projektet. Idag kan vi dock finna en stor del av mjukvara vilken inte har någon koppling till GNU projektet men ändå är distribuerad under GPL, ett bra exempel på detta är Linux kärnan. GPL licensen var utformad så att den skulle främja produktionen av öppen programvara och på grund av detta så är det strikt förbjudet att modifiera program under denna licens så att det kan integreras med proprietär programvara (Gonzalez-Barahona, 2000).

GPL är baserad på den internationella lagstiftningen om upphovsrätt, vilken försäkrar dess upprätthållande. Huvudkaraktäristika för GPL är att det tillåter distribution, modifikation av källkod och integration med annan programvara. Modifiering tillåts utan några restriktioner om reviderad programvara också täcks av GPL. Integration är endast möjligt om de andra programvarorna också täcks av GPL. Detta är dock inte fallet för LGPL (GNU Lesser General Public License) även det använt i GNU projektet, vilket tillåter integration med nästan vilken programvara som helst, inklusive proprietär programvara (Gonzalez-Barahona, 2000).

4.1.5 För och nackdelar med öppen programvara

Den största fördelen med öppen programvara inom offentlig sektor och även generellt sätt är de ekonomiska besparingar som kan göras vid användning av denna typ av programvara. Öppen programvara har nämligen inte de omfattande licenskostnaderna som proprietära program vanligtvis har vid användning på flera datorer. Detta brukar vara det största argumentet för användning av öppen programvara och samtidigt den största ekonomiska besparingen (Kihlström & Kihlström, 2002).

För en utvecklare finns det andra fördelar som till exempel möjligheten att modifiera programmet och återanvända källkod från tidigare öppen-källkodsprojekt. Kodgranskning sker kostnadsfritt och eftersom öppna programvaror distribueras fritt möjliggörs också granskning av andra programmerare än de som varit med och utvecklat källkoden. Detta leder i sin tur till att fel både kan hittas och åtgärdas snabbare, vilket medför att säkerheten och kvaliteten höjs på öppna programvaror (Kihlström & Kihlström, 2002).

Förutom att en investering i öppen programvara kan medföra kostnadsbesparingar, är det även en försäkring mot leverantörsbortfall. Vid inköp av proprietär programvara medföljer risker, som till exempel om en leverantör går i konkurs eller byter inriktning, kan kunden bli utan underhåll, uppgraderingar eller modifieringar av programvaran, som tidigare anskaffats. Ett oberoende skapas vid användning av öppen programvara eftersom förändringar och uppgraderingar kan göras vid behov då koden är tillgänglig för användaren (Kihlström & Kihlström, 2002).

Med andra ord behöver inte användaren vända sig till någon specifik leverantör och be om lov, så länge denne följer de regler som står skrivna i licensavtalet för den öppna produkten. Praktiskt sett har användaren frihet att göra vad denne önskar med

den öppna programvaran, men vidaredistribution får ej ske utan att göra den omarbetade källkoden tillgänglig för allmänheten, kostnadsfritt.

Den största nackdelen med öppen programvara är bisten på tillförlitlig dokumentation. Dokumentation kan vara svår att organisera och underhålla då arbetet kan ske distribuerat globalt, vilket kan leda inaktuell dokumentation. Krav på användning av metoder och modeller, kravspecifikationer eller systemdokumentation vid utvecklandet finns inte, vilket också resulterar i att tillförlitligheten för arbetet minskar (Kihlström & Kihlström, 2002).

Utöver detta finns det fortfarande svårigheter med att få tag i personal med rätt kunskaper för en del specifika öppna program. Antalet leverantörer som håller i utbildningar vad gäller installation, konfiguration och användning av dessa system är ännu begränsat. På samma sätt är tillgången till support och garantiavtal från leverantörer avsevärt liten. En del anger att öppen programvara har sämre säkerhet än proprietär, men detta är oftast felaktigt. Självklart kan det finnas brister i säkerheten, men dessa är dock inte mer omfattande än de brister som finns i programvaror som är proprietära. Fel i säkerheten hos öppna programvaror brukar dessutom korrigeras mycket snabbare (Kihlström & Kihlström, 2002).

Ett problem med användningen av öppen programvara är att det kan vara svårt att göra storskaliga användartest för att förbättra användarvänligheten, så länge det inte är ett kommersiellt bolag med stora resurser. Öppen programvara har ingen bestämd standard vad gäller gränssnittet till skillnad från proprietär programvara, vilket kan medföra problem för enhetligheten hos programvarorna (Broersma, 2002).

4.2 IT-investeringar

4.2.1 Vad är IT-investerings och beslutsprocess modeller/metoder?

Enligt Schniederjans, Hamaker & Schniederjans (2004) finns det flera olika definitioner av informationsteknologiinvesteringar eller IT-investeringar. En teori är att IT-investering är en term som kan appliceras vid investering av utrustning, applikationer, tjänster och grundläggande teknologier. Andra anger att IT-investeringar är de kostnader associerade med inköp av datorer, kommunikation, nätverk och personal som kan förvalta och driva ett management information system (MIS).

”Ett Management Information System, är ett företagsledningssystem och ett beslutsstödsystem. De är datorbaserade system för hantering och bearbetande av information. Avsikten med detta är att ge olika nivåer i en företagsledning den information som behövs för att kunna leda verksamheten och fatta beslut.”

(Thorell, 2005)

Definitionen som författarna Schniederjans et al. (2004) använder sig av är en där samtliga komponenter ingår som utgör ett MIS. Alla MIS är en samling av fyra primära komponenter: personal, applikations mjukvara, system mjukvara och hårdvara. Ett MIS består av personal som förvaltar och driver företagets Informationsteknologi. Personalen kan bestå av användare som behöver teknologisk utbildning och kan där-

för också räknas som en IT-investering, personalen med teknisk kompetens vilken hanterar input/output funktioner i systemet och driver de operativa data systemen och deras chefer. Andra komponenter inkluderar applikationsmjukvara som till exempel programmerings kod, assembly kod, C++ etc. och system mjukvara som till exempel operativsystem, tolkare, kompilator, program för att hantera data etc. Kärnan i denna modell består av IT hårdvarans drivande interaktiva komponenter som till exempel datorer, hårddiskar och system, kommunikationssystem, nätverkssystem etc.

Fokus ligger på relationen mellan resurser som är allokerade till de individuella komponenterna såväl som det samlade systemet som utgör MISet. Definitionen av IT-investering kan definieras som investerings beslut för allokering av alla kategorier (mänskliga, monetära, fysiska etc.) av resurser till ett MIS (Schniederjans et al., 2004).

Det primära området för vilka IT-investeringsmodeller/metoder är känsliga är relaterat med risk. Risk kan delas upp i två klasser; den första är *fysisk* risk vilken avser sårbarheten hos datorers hårdvara och mjukvara, och data för sabotage, stöld; mjukvarusårbarhet som piratkopiering och borttagning. Den andra är *operationell* risk vilken innebär misslyckande i att uppnå förutspådda fördelar eller kostnads reduceringar: misslyckande av implementation för att uppnå en önskad tidsram; slut-användar mottstånd; ett systems oförmåga att stödja organisation eller dess tillväxt under tiden; och inkompatibilitet problem som senare utvecklas (Schniederjans et al., 2004).

Genom att välja den bästa modellen eller modeller kan risken minimeras avsevärt. Det är därför viktigt att studera IT-investerings och beslutstagande metodologier (Schniederjans et al., 2004).

4.2.2 Modeller för IT-investeringar

Idag finns det relativt få modeller att tillgå trots det ökade intresset för detta område. De modeller som finns är inte tillräckligt täckande vad gäller stödjandet av arbetet från start till slut. Lundberg (2004) presenterar en ny modell baserad på tidigare mer traditionella modeller vilken heter 5-modellen. Något de befintliga modellerna har gemensamt är att samtliga på olika sätt försöker finna sambanden mellan IT-investeringar och företagets resultat och framgång. Författaren anger att de modellerna fokuserar mer på ekonomin än på IT.

Det som Lundberg anser fallerar med arbetet med de befintliga modellerna är att de prognoser som gjorts med hjälp av nyttokalkyler i inledningsskedet av projektet inte stämmer överens med slutresultatet. Detta beror på bristande verifiering och uppföljning av kalkyler under rådande arbetsförlopp.

Krav som bör ställas på en modell för IT och affärsnytta är bland annat att den ska vara ett stöd vid värderingen av affärsnyttan. Den ska även stödja både hårda ekonomiska värden samt de mjukare värden som finns i verksamheten. Modellen ska vara ett stöd genom att arbetet skall kunna följas upp och kunna hämta hem den planerade affärsnyttan (Lundberg, 2004)

Egenskaper hos 5-modellen är:

- Den ser till helhetsperspektivet på hur nytta ska realiseras, inte bara beräknas.
- Både kvalitet och kvantitet hanteras av modellen. Hårda ekonomiska parametrar kan blandas med mjuka faktorer som mäts via olika nyckeltal i verksamheten.
- Modellen innehåller verktyg som ökar spårbarheten från förändringar till verklig nytta i verksamheten.
- Modellen tar hänsyn till de totala kostnaderna för en satsning så att inte nytan går förlorad i oväntade eller dolda kostnader.
- Modellen tar även hänsyn till hur nyttan utvecklas med tiden och vad som kan göras för att förlänga nyttans livslängd.

4.2.3 Vad är affärsnytta?

Indelningen av affärsnytta kan göras i två kategorier, nytta i form av minskade IT-kostnader och nytta i verksamheten som sådan. Delarna är båda viktiga men den IT-mässiga nyttan är dock sällan så stor som den verksamhetsmässiga nyttan (Lundberg, 2004).

Med affärsnytta avses:

- Kostnadsbesparingar, vilket avser kostnaden för att driva en verksamhet före införandet av nytt IT minus kostnaden efteråt.
- Intäktsökning, där intäkterna före införandet av IT minus intäkterna före beaktas.
- Kvalitativ nytta, vilket visar på kvalitativa förbättringar i de nyckeltal som används i verksamheten (kundnöjdhet, personalomsättning, trivsel etc.).
- IT-nytta kan beskrivas som kostnader för IT före förändringen minus kostanden efteråt.

(Lundberg, 2004)

Genom en jämförelse av dessa poster kan en uppskattning av netto nyttan göras. Andra kostnader som är av vikt att beakta är förändringskostnaderna som till exempel, projektkostnader, IT-kostnader och omställningskostnader (Lundberg, 2004).

4.3 Tidigare studier/utredningar

4.3.1 IT-användning och strategier vid Universitet och Högskolor

I en utredning gjord 1995 av Kulturdepartementet på begäran av regeringen presenteras information rörande bland annat universitet och högskolors IT-användning och strategier i Sverige. Där står det att dessa verk har kommit långt i implementering och tillämpning av IT, problemet är dock att endast ett fåtal har upprättat en IT-strategi. Enligt undersökningen så är de universitet och högskolor som inte har en strategi, i arbete med att ta fram en. Samtliga högskolor och universitet med IT-strategier anser att det är av vikt att informationsteknik används inom forskning och utbildning vid

landets universitet och högskolor. Syftet med strategierna enligt institutionerna är att öka IT-användningen och på så sätt rationalisera och effektivisera verksamheten (Håkansson, 1996).

De gemensamma målen som institutionerna har för IT-strategierna är att studenterna skall ha möjlighet att använda IT i utbildningen och att det skall vara ett självklart redskap i den dagliga verksamheten. Det skall även användas vid informationsspridning och publicering av akademiskt material, samt att IT skall användas direkt i utbildningen, vid till exempel distansundervisning. Hur dessa målsättningar skall uppnås finns formulerat i form av konkreta förslag i IT-strategierna. Ett förslag är till exempel att den tekniska infrastrukturen byggs ut så att studenter får tillgång till nät-tjänster i bostaden och att e-post och konferenssystem blir naturliga sätt att kommunicera på (Håkansson, 1996).

Tillskillnad från andra institutioner lägger den akademiska världen stor vikt vid Internet. Informationen på högskolor och universitet presenteras vanligtvis på Internet, vanlig post ersätts med e-post, arkiv och källmaterial skall finnas tillgängligt digitalt etc. Utbildning är även något som anses vara av stor vikt eftersom att det är essentiellt för studenter och personal att lära sig använda den nya tekniken för att datorer skall kunna bli ett självklart verktyg i utbildningen. Idag har ett flertal universitet och högskolor etablerat speciella IT-råd som skall ansvara för dessa frågor (Håkansson, 1996).

4.3.2 Modell för utvärdering av öppen och proprietär programvara

I en rapport från Statskontoret (2003) presenteras en modell för evaluering av öppen respektive proprietär programvara. Två aspekter av investeringar tas upp i denna modell en som berör användningsrätten och en annan som berör underhållet av produkten. I den initiala investeringen innefattas den största delen av kostnaderna för användningsrättigheterna till skillnad från underhållsperspektivet där de ekonomiska faktorerna avser en längre tidsperiod med större osäkerhet (Modellen framgår av bilaga 2). Denna modell har vi baserat en av de frågor som finns med i frågeformuläret på. Vi har utvecklat modellen genom att komplettera med en del kriterier som vi anser vara av vikt vid utvärdering av öppen och proprietär programvara.

4.3.3 Utredningar om öppen/proprietär programvara i Sverige

I Sverige har ytterst få undersökningar gjorts vad gäller öppna respektive slutna miljöer. Idag finns det dock uppgifter rörande användningen av öppen programvara inom den svenska förvaltningen, till skillnad från 2003 då Statskontorets rapport presenterades. Statskontorets ramavtal utnyttjades redan då i stor utsträckning vid införskaffning av både hårdvara och mjukvaruapplikationer och används än idag. Detta avtal är avsett för nyttjande av myndigheter, kommuner och landsting. Microsofts produkter är dominerande på arbetsplatserna för operativsystem och kontorsprogram. Under 2002 omsatte PC-avtalen ca 3 miljarder kronor, vilket innefattar arbetsplatsutrustning såväl som hem-pc. För att leverantörerna skall erhålla rabatterade priser måste datorerna vara försedda med ett förinstallerat operativsystem där kostnaden för ett uppgår till ca: 1500 kronor (Statskontoret, 2003).

I rapporten från Statskontoret redovisas också uppgifter om besparingar på ca 5 miljarder SEK per år på kort sikt inom förvaltningen i Danmark vid övergång till öppen programvara, vilket avser både besparingar för datorer och programvara. Med ca 450 000 arbetsplatsdatorer i Danmark kan besparingarna i Sverige bli ännu större då det finns ca 700 000 arbetsdatorer där. I en utredning utförd av Arbetsmarknadsverket (AMS) redovisas vad en övergång från Microsofts kontorsprogram Office till *StarOffice* skulle innebära ekonomiskt. Resultatet av undersökningen visar att licenskostnaderna skulle kunna reduceras avsevärt vid en övergång till StarOffice till skillnad från uppgradering till en ny version av Microsoft Office. Uppgraderingen till en ny version av MS Office uppskattas kosta 25 miljoner SEK, medan licenskostnaderna för StarOffice beräknas uppgå till sju miljoner, där interna kostnader för marknadsföring av produkten är inräknade (Statskontoret, 2003).

4.4 Diffusionsteori

4.4.1 Vad är diffusion?

Själv ordet har sitt ursprung i det latinska ordet *diffusio* som kommer av *diffundo* och kan på svenska översättas till att sprida (sig). Begreppet diffusion lanserades på 1870-talet som en av de möjliga förklaringarna till förekomsten av likartade kulturelement i skilda geografiska områden (Nationalencyklopedin, 2005). I boken "Diffusion of innovation" definierar författaren Rogers (1995) diffusion på följande sätt;

"Diffusion is the process by which an innovation is communicated through certain channels over time among the members of a social system."

Definitionen innehåller vissa oklarheter men Rogers (1995) anger vidare att förklaringen ligger hos de fyra huvudelementen; innovation, kommunikations kanaler, tid och socialt system. I varje forskningsstudie som genomförts berörande diffusion kan dessa element identifieras.

4.4.2 Innovation

Innovation är en idé eller ett objekt som uppfattas som något nytt av individen. Tiden som den nya innovationen har funnits historiskt sett räknas inte förrän individen blivit introducerad för den nya innovationen. Det behöver inte heller vara en ny idé eller innovation som individen kommer i kontakt med, utan det är först när individen antar en åsikt om nyheten som det räknas som en innovation. Rogers (1995) har tagit fram en modell som är erkänd hos forskare inom diffusionsteorin samt forskningsfakulteter i allmänhet. Många forskningsavhandlingar baserar sig på hur snabbt en individ tar till sig en ny innovation. Modellen som Rogers tagit fram illustrerar just hur detta går till och hur många procent av individerna som finns i de olika kategorierna han skapat.

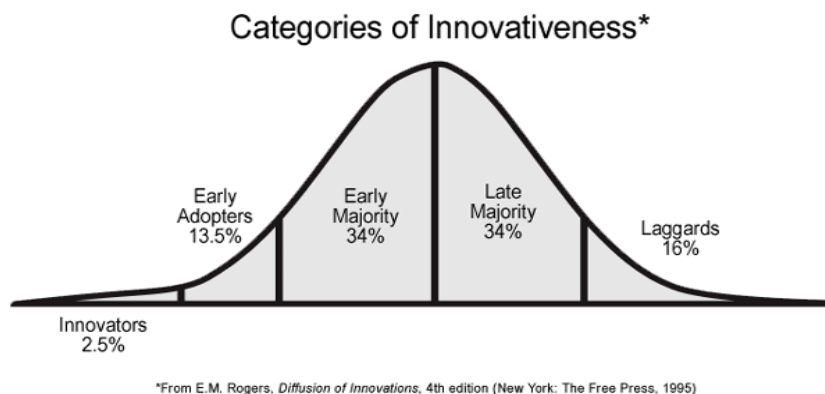


Figure 1 Adopter categorization based on innovativeness (Rogers, 1995)

Rogers (1995) föreslår att normalkurvan fördelas som följer;

- Första 2.5% kallas "innovators" (innovatörer)
- Nästa 13.5% kallas "early adopters" (tidiga accepterande)
- Nästa 34% kallas the "early majority" (tidig majoritet)
- Nästa 34% kallas the "late majority" (sen majoritet)
- Sista 16% kallas "laggards" (efterslänrare)

(Rogers, 1995 s.11)

Denna uppdelning grundar sig på extensiv forskning gjord inför boken "Diffusion of innovation" och Rogers (2004) menar att de två första grupperna skall fokuseras på när en teknologisk innovation skall introduceras, och få en skjuts framåt.

Dessa två grupper studeras ytterligare i detalj och karaktäriseras genom observationer som gjorts. Innovatörerna kännetecknas som entreprenörer, med fokus på risktagande både ekonomiskt och personligt. Typiskt för en innovatör är också en hög tålighet beträffande osäkra förhållanden när de accepterar en ny innovation. De måste vara förberedda på att göra misstag och kunna hantera detta. Innovatörerna kommunicerar ofta med andra innovatörer trots geografiska avstånd, och detta skapar ett stort nätverk med likasinnade. Detta gör också att innovatörerna inte är respekterade av andra medlemmar i det sociala samhället, eftersom de håller sig utanför det "normala". Innovatörernas betydelse är stor och det är av vikt att framhålla deras funktion (Rogers,1995).

Den andra gruppen som kallas "tidiga accepterande" är betydligt mer integrerade i det sociala samhället jämfört med innovatörerna. Denna kategori har den största påverkan på "gemene mans" åsikter. Andra grupper vänder sig gärna till "tidigt accepterande" gruppen för att få råd om en ny innovation innan beslut fattas. Denna grupp är också eftertraktad av diverse intressenter, bland andra förändringsagenter, eftersom de ofta accepterar en innovation snabbare än de andra grupperna. "Tidigt accepterande" gruppen har ofta en central position i organisationer och hon eller han måste ta

många och svåra beslut vad beträffar innovationer. Detta gör denna grupp till en respekterad och inflytelserik grupp eftersom deras beslut är genomtänkta och därför kan gynna de flesta andra grupper (Rogers,1995).

4.4.3 Kommunikationskanaler

Kommunikation definieras som ”en process där deltagare skapar och delar information med varandra i syfte att få en gemensam förståelse” (Rogers, 1995). Diffusion är en speciell kommunikation där informationen som delas härrör en ny idé. För att simplificera processen kan man säga att på den mest elementära nivån finns det (1) en innovation, (2) en individ eller en annan organisation som har kunskap om innovationen, eller erfarenheten av innovationen, (3) en annan individ eller annan organisation som inte har erfarenhet av innovationen, och (4) kommunikationskanaler (Rogers, 1995).

En kommunikationskanal är ett medel för att få sitt meddelande kommunicerat från en individ till en annan. Massmediakanaler är ofta de mest effektiva kanalerna när en stor folkmassa skall informeras, och involverar ett massmedium som television, tidningar eller radio, med flera. Det är också effektivt att använda sig av interpersonella kanaler för att övertyga individen ytterligare. Detta är speciellt verkningsbart om båda individerna har samma socioekonomiska status och/eller samma utbildning. Utredningar visar att de flesta individerna inte evaluerar en ny innovation efter resultat som forskning kommit fram till, även fast detta inte är helt utan relevans, utan istället litar på andra individer i omgivningen som tidigare har accepterat innovationen. Detta betyder att ju mer homogena individerna är desto smidigare accepteras innovationen hos alla i gruppen (detta kan jämföras med gruppen ”tidiga accepterande”) (Rogers,1995).

4.4.4 Tid

Tid är det tredje elementet i diffusionsprocessen. Denna variabel inkluderas sällan inom andra beteendevetenskapliga ämnen, orsak till detta är att tid ibland kan vara svårt att mäta. Tidsaspekten är en viktig del av diffusion och (1) involveras genast i innovationsbeslutsfattandeprocessen som en individ går igenom när den får erfarenhet av en ny innovation och därefter får en uppfattning om innovationen. (2) graden av innovationsanpassning, relativt tidigt eller sent, jämfört med andra individer i samma sociala system, (3) och i vilken takt medlemmarna i det sociala systemet accepterar innovationen inom en given tidsram (Rogers,1995).

4.4.5 Socialt system

Ett socialt system består av delar som är interrelaterade till varandra och som tillsammans engagerar sig för att lösa problem och uppnå gemensamma mål. Medlemmarna i ett socialt system kan vara individer, informella grupper, organisationer och/eller undersystem (Rogers,1995).

Det sociala systemet innehåller gränser inom vilka diffusion sker. Vilka gränserna är beror bland annat på hur den sociala konstruktionen av systemet ser ut samt hur in-

novationsbeslutsfattande går till. Dessa ämnen involverar hur förhållanden mellan det sociala systemet och diffusionsprocessen ser ut (Rogers,1995).

Alla individer är inte lika i ett socialt system och därför finns det en primär struktur som ska hjälpa till att ge regularitet och stabilitet, detta gör det möjligt att förutse gruppens beteende. Denna struktur kallas den formella strukturen och är den vanliga beslutsgången, det vill säga en beslutsfattare meddelar en underordnad om vad som ska göras, och förväntar sig sedan att detta blir utfört. Det finns också en annan struktur som kallas för den informella strukturen, vilket bestämmer vem som kommunicerar med vem och under vilka förhållanden. Detta definieras som "kommunikationsstruktur" och fungerar som ett differentierade element eftersom detta ibland är överordnat den formella strukturen (Rogers,1995).

Dessa fyra ovanstående begrepp; innovation, kommunikationskanaler, tid och socialt system är tillsammans grunden för vilka kriterier som spelar in när en individ ska ta beslut om att acceptera en teknisk innovation (Rogers,1995).

4.4.6 Innovation inom en organisation

Den innovationsbeslutsfattandeprocessen fokuseras huvudsakligen på individen, men många innovationsbeslut görs av organisationer istället för individuellt. Rogers (1995) skiljer på individ- och organisationsnivå i diffusionsteorin och menar att det ofta är svårt för en individ att acceptera en innovation innan organisationen gjort detsamma. Innovationsprocessen ser annorlunda ut för organisationer och skiljer sig från den så kallade innovationsbeslutsfattandeprocessen som individen genomgår innan beslut. Inom organisationen är diffusionsprocessen mycket mer komplicerad och involverar olika individer vilka antar olika roller. Det finns tre typer av organisatoriska innovationsbeslutsfattande processer;

- *Valfria innovationsbeslut*

Val att acceptera, eller ej acceptera, en ny teknisk innovation görs oberoende av hur andra medlemmar i det sociala systemet agerar.

- *Kollektiva innovationsbeslut*

Val att acceptera, eller ej acceptera, en ny teknisk innovation görs på grunder av att alla i det sociala systemet är överens och det råder en samstämmighet.

- *Auktoriserade innovationsbeslut*

Val att acceptera, eller ej acceptera, en ny teknisk innovation görs av relativt få individer som besitter makten, statusen eller den tekniska expertisen.

Det finns ytterligare en organisatorisk innovationsbeslutsfattande process som kallas för den *beroende innovationsbeslutsprocessen* som betyder att ett beslut tas först efter ett annat innovationsbeslut har tagits. Det är en kombination av någon av de ovanstående processerna. Till exempel beslutar ledningsgruppen på ett systemutvecklingsföretag sig för att införskaffa ett nytt tidsrapporteringsystem, men det är valfritt för personalen att använda systemet under en övergångsperiod av två månader. Detta är

först ett auktoriserat innovationsbeslut som sedan följs av ett valfritt innovationsbeslut. Dessa processer är användbara för att definiera vilket sorts beslut som tas och hur det går till (Rogers,1995).

4.4.7 Ytterligare begrepp beträffande acceptans

Enligt Rogers (1995) finns det även andra aspekter att tänka på förutom de ovan nämnda begreppen. Moore och Benbasat (1991) utvecklar Rogers koncept ytterligare i artikeln ”Development of an Instrument to Measure the Perceptions of Adoption and Information Technology Innovation”. De definierar begreppen och sammanfattar diffusionsteorin med hjälp av frågor och en konceptualiserad modell. Syftet med deras rapport är att ta fram ett verktyg för att studera den initiala diffusionsprocessen inom IT-området.

Modellen bygger på sex begrepp som Rogers (1995) tagit fram samt ett sjunde begrepp som enbart är framtaget av Moores och Benbasat (1991);

- **Relativa fördelar:** I vilken utsträckning innovationen uppfattas vara bättre än föregångare.
- **Image:** Graden av hur en innovation kan upphöja imagen eller statusen i ett socialt system.
- **Kompabilitet:** Om innovationen uppfattas vara konsistent med de existerande värderingar, behov, samt tidigare erfarenheter som potentiella kunder haft.
- **Komplexitet:** Vilken svårighetsgrad den nya innovationen uppfattas ha.
- **Observabilitet:** Hur pass överblickbara resultaten är som innovationen medför.
- **Trialabilitet:** Hur mycket det går att experimentera med innovationen innan införande.

Ett sjätte attribut som Moore och Benbasat lagt till i sin studie är

- **Frivillighet:** Hur frivillig innovationen känns för användaren.

Begrepp som Moore och Benbasat (1991) utvecklat, i kombination med Rogers individuella och organisatoriska innovationsbeslutprocesser, ger en bild av vilka villkor som ska vara uppfyllda för att en teknisk innovation ska bli accepterad av både individer såväl som organisationen.

5 Resultat/Analys

I följande kapitel kommer vi att redovisa det mest relevanta resultatet från den empiriska undersökningen. Det sammanställda resultatet för de tre läroanstalterna kommer tillsammans med relevant teori från ramverket analyseras och utvärderas.

5.1 Öppen programvara

5.1.1 Definition av öppen programvara

De personer vi intervjuat vid respektive universitet/högskola skiljer sig åt vad gäller deras uppfattning om och definition av öppen programvara, men de är alla entusiastiska då vi talar om fenomenet öppen programvara. Majoriteten av de tillfrågade är överens om att denna typ av programvara inte nödvändigtvis är gratis eftersom att den ofta är skyddad av licenser av olika slag. Detta överensstämmer med den definition som anges i Statskontorets rapport från 2004, vilken innebär att denna typ av programvara i de flesta fall inte är licensfri och därmed inte nödvändigtvis gratis. Yngvesson (2005) på Internationella Handelshögskolan i Jönköping menar att denna programvara kan anses vara gratis eftersom att den inte har några initiala investeringskostnader, han syftar då främst till operativsystemet Linux. Att öppen programvara associeras med operativsystemet Linux är ingen överraskning eftersom att det är en av det mest kända produkterna bland öppna programvaror. Yngvesson avser endast en produkt i sin definition av öppen programvara och därför är det svårt att jämföra hans definition med de övriga respondenternas uppfattningar.

Henriksson (2005) vid Högskolan i Jönköping associerar öppen programvara med ett kollektivt ägande och som drivs av användarföreningar, han tillägger också att det inte finns någon förvaltningsorganisation. Till skillnad från Henriksson (2005) lägger Bengtsson (2005) vid Uppsala universitet, i sin definition större vikt vid licenser för öppen programvara, eftersom han anser att det är dem som bidrar till den fortsatta utvecklingen av programvaran. De flesta licenser för öppen programvara kräver att vidareutvecklingar görs tillgängliga för allmänheten, vilket möjliggör en fortsatt utveckling av programvaran (Wikipedia, 2005). För att den öppna programvaran ska kunna vidareutvecklas är det av vikt att upphovsrättsinnehavaren specificerar licensvillkor för användandet av programvaran, om detta inte görs är det upphovsrättslagen som råder vilken begränsar en användares friheter. Att släppa en programvara fri under en licens blir därför nödvändigt för att ge användaren mer friheter (Susning, 2004).

Nejdeby (2005) talar hellre om öppna standarder än öppen programvara, som kan definieras som en standard vilken är tillgänglig för allmänheten. Samma standard krävs för att uppnå kompatibilitet mellan produkter från olika tillverkare. Eftersom att öppna standarder skapar möjligheter för vilken tillverkare som helst att producera en produkt som är kompatibel, är det önskvärt med system som baseras på öppna standarder (Andrén, 2003). Öppen programvara har dessvärre ingen bestämd standard vad gäller gränssnittet, till skillnad från proprietär programvara, vilket kan medföra problem för enhetligheten hos programvarorna och kompatibiliteten med andra produk-

ter (Broersma, 2002). Varför Nejdeby (2005) hellre väljer att tala om öppna standarder kan bero på att han anser att öppen programvara innebär kompatibilitetsproblem. Genom att ha standarder som är enhetliga skulle integration mellan öppna och proprietära program, men också öppna program sinsemellan, underlätta. Nejdeby (2005) refererar också till ”Community Source modellen” i sin definition av öppen programvara, vilken skiljer sig från öppen programvara modellen eftersom att den kräver kompatibilitet bland tidigare utvecklade versioner av programvara, proprietära såväl som öppna (Gabriel & William, 2005).

5.1.2 Erfarenheter av öppen programvara

Flera års erfarenhet av öppen programvara finns inom samtliga tillfrågade högskolor och universitet. På Linköpings universitet fick denna typ av programvara inte genomslagskraft förrän på 90-talet, detta trots erfarenhet av öppen programvara sedan universitetet grundades 1975. 90-talets början var en milstolpe i utvecklingen av öppen programvara. Internet och USENET möjliggjorde integration av olika användargrupperns arbeten vilket ledde till att kompletta miljöer kunde byggas ovanpå UNIX med nyttjandet av öppen programvara. Det var också under denna tidsperiod som Linus Torvald utvecklade de första versionerna av Linuxkärnan (Gonzalez-Barahona, 2000). Vi tror att förändringar i utvecklingen av öppen programvara på andra sidan Atlanten kan ha varit orsaken till det ökade intresset för öppen programvara på Linköpings universitet. Detta eftersom att intresset ökade i samband med de revolutionära framstegen som gjordes inom öppen programvaruområdet.

5.1.3 Användning av öppen programvara

Samtliga universitet och högskolor använder öppen programvara, men det varierar mellan läroanstalterna och en del använder det i större utsträckning än andra. På högskolan i Jönköping används cirka 10-15 applikationer i driftmiljön, dessa är bland andra, Apache (webbserver), Php (skriptspråk), My-SQL (databas) och Novell Netware tillbehörsprogram. I utbildningsmiljön finns Linux operativsystem installerat i enskilda datasalar. På Uppsala används öppen programvara i olika former, kontorsprogram som Open Office, serveroperativ som Linux och databaser som My-SQL. Portalen U-portal och lärandeplattformen Sakai är fortfarande under utveckling.

På universitetet i Linköping används Apache och Linux inom de infrastrukturella tjänsterna och inom universitetets gemensamma IT-tjänster används bland annat desktop management system för Linux, Sun Solaris och Staroffice. Samtliga läroanstalter nyttjar Apache, My-SQL och Linux i någon utsträckning. Inom många branscher är det bästa alternativet redan öppen programvara, ett utomstående exempel är Apache webbserver med över 50 % av marknaden (Gonzalez-Barahona, 2000). Linux som är baserad på egenutvecklad källkod, är idag det mest kända operativ systemet bland öppna alternativ. Generellt sett ser det ut som att läroanstalterna brukar de mest etablerade öppna programvarorna på marknaden, vilket tyder på att denna typ av programvara ses som ett erkänt alternativ till proprietära produkter.

5.1.4 Varför används/används inte öppna program?

Statskontorets rapport (2004) visar att öppen programvara främst nyttjats inom den offentliga förvaltningen för besparingsskäl. Kihlström & Kihlström (2002) skriver i likhet med Statskontoret att den största fördelen med denna typ av programvara inom offentlig sektor är de ekonomiska besparingar som kan göras vid dess användning. Samtliga universitet/högskolor som deltagit i undersökningen anger att investeringar i öppen programvara inte beror på besparingsskäl, tvärtom är denna typ av programvara ett dyrare alternativ för läroanstalterna. Den största anledningen till att Internationella Handelshögskolan i Jönköping väljer bort öppen programvara är enligt Yngvesson (2005) de förmånliga avtalen som Microsoft erbjuder. Yngvesson (2005) menar att de priser som Microsoft erbjuder är så låga att en övergång till annan leverantör eller annan typ av mjukvara inte är ett alternativ.

Henriksson(2005) instämmer med Yngvesson (2005) och anger också att det begränsande användandet av öppen programvara beror på att ett nyttjande i större utsträckning skulle driva upp kostnaderna för högskolan. Vidare påpekar han att de nuvarande avtal som de har med Microsoft är så fördelaktiga att en övergång inte är aktuell av ekonomiska skäl.

Till skillnad från Henriksson(2005) och Yngvesson(2005) anser Nejdeby (2005) på Linköpings universitet att de låga licenskostnaderna för öppen programvara inte är av någon större betydelse för universitet och högskolor.

Microsofts produkter dominerar inom verksamheterna, vilket beror på de förmånliga avtalen som Microsoft erbjuder läroanstalterna. De priser som Microsoft erbjuder är så låga att det enligt vår åsikt mer kan ses som en symbolisk summa. Microsoft har strategiskt riktat in sig på utbildningsanstalter för att sälja in sina produkter, och trots låga priser är det Microsoft som på lång sikt tjänar mest på avtalet (Yngvesson, 2005). Strategin är mycket listig eftersom att de produkter som studenter använder under sin utbildning troligtvis är de som de kommer att använda efteråt.

Yngvesson (2005) anser dock att det är essentiellt att regelbundet utvärdera de alternativ som finns på marknaden för att följa med i den snabba tekniska utvecklingen, och kunna erbjuda studenter och anställda de bästa alternativen efter skolans förmåga. Det som Yngvesson påpekar är av vikt, då vi anser att lärosätena ska vara neutrala institutioner i samhället och ansvara för att bidra till diversifierade kunskaper både bland studenter och anställda vad gäller olika mjukvaruprodukter på marknaden. Att som läroanstalt endast förespråka proprietära alternativ kan tolkas som ett politiskt ställningstagande, där en viss kategori av tillverkare gynnas. Vi tror därför att det är viktigt att även framhålla produkter av öppen karaktär.

5.1.5 För- och nackdelar med öppna program

En stor nackdel men också fördel som Nejdeby (2005) nämner är att öppen programvara skapar oberoende som leder till större frihet men samtidigt mindre trygghet. Kihlström & Kihlström (2002) anger att risker medföljer vid inköp av proprietär programvara, och ger som exempel leverantörer som går i konkurs eller byter inriktning. Detta resulterar i att kunden blir utan uppgraderingar eller modifieringar av

programvaran som tidigare anskaffats. Med andra ord skapas ett beroende av leverantören. Öppen programvara är en försäkring mot leverantörsbortfall och skapar ett oberoende, då förändringar och uppgraderingar kan göras vid behov, eftersom koden är tillgänglig för användaren (Kihlström & Kihlström, 2002). Vi anser dock att detta kan diskuteras, eftersom ett proprietärt alternativ samtidigt ger trygghet i form av den support som ofta kommer med inköpet av produkten, medan öppna produkter lägger över hela ansvaret på användaren.

Det finns en nackdel som Henriksson (2005) påpekar med öppen programvara och det är avsaknaden av applikationer som motsvarar de proprietära alternativen vilka finns på marknaden idag. Detta leder till att det blir svårt att använda öppen programvara fullt ut. Henriksson (2005) anser dock att öppen programvara har många fördelar, men att tekniken idag inte är tillräckligt mogen för ett mer omfattande användande inom organisationen. Några nackdelar Bengtsson (2005) anser finns med öppen programvara är till exempel att det saknas viss kompetens på området och att det inte finns en tydlig supportfunktion, men han påpekar att detta successivt håller på att förändras. Kihlström & Kihlström (2002) menar också att det fortfarande finns svårigheter med att få tag i personal med rätt kunskaper för en del specifika öppna program. Orsaken till detta är att antalet leverantörer som håller i utbildningar vad gäller installation, konfiguration och användning av dessa system, ännu är begränsat. På samma sätt är tillgången till support och garantiavtal från leverantörer avsevärt liten (Kihlström & Kihlström, 2002). Vi anser att läroanstalterna bör ha en större roll vad gäller utbildning i öppna program, för att öka kompetensen inom området. Genom att göra detta kanske intresset för denna typ av programvara skulle växa inom både skolväsende och näringsliv. Följaktligen skulle det troligtvis också underlätta för företagen att finna kompetent personal.

Bengtsson (2005) anger också att öppen programvara är svårt att använda för den oerfarne. Detta problem beskriver även Nejdeby (2005) som förklarar att förutsättningarna för användandet av öppen programvara givetvis har förändrats över tiden, men att utvecklingen av den typen av programvara alltjämt sker av unga killar, tekniker. Detta leder till problem då programmen ska lanseras för "vanliga" användare, som kräver användarvänliga program. Broersma (2002) påpekar också detta problem, och anger att det beror på de svårigheter som finns med att göra storskaliga användartest för att förbättra användarvänligheten, så länge det inte är ett kommersiellt bolag med stora resurser. Vi tror att sanningen ligger någonstans mitt i mellan, det beror säkert på avsaknad av storskaliga användartester till viss del, men också att ett stort antal mindre projekt är för teknikstyrda, då det ofta utvecklas av tekniker allena.

Fördelar med öppen programvara enligt Bengtsson (2005) är att den är mycket säkerhetsmedveten, och installationer och uppdateringar som tidigare har varit krångligt är mycket enklare utformade i nyare distributioner. Vad gäller säkerhetsaspekten av öppen programvara förklarar Kihlström & Kihlström (2002) att säkerheten och kvaliteten höjs på öppna programvaror delvis för att utvecklare har möjligheter till att modifiera programmet och återanvända källkod från tidigare öppen-källkodsprojekt, men också för att kodgranskning sker kostnadsfritt av andra programmerare än de som varit med och utvecklat källkoden. Detta leder i sin tur till att fel både kan hittas och åtgärdas snabbare. Kihlström & Kihlström (2002) anger att en del påstår att öppen

programvara har sämre säkerhet än proprietär, men hävdar att detta är oftast osanningar. De anser att det självklart kan det finnas brister i säkerheten, men dessa är dock inte mer omfattande än de brister som finns i programvaror som är proprietära. Fel i säkerheten hos öppna programvaror brukar dessutom korrigeras mycket snabbare (Kihlström & Kihlström, 2002). Beträffande säkerheten anger Henriksson (2005) att skillnaderna är marginella vid jämförelse av öppen och proprietär programvara. Den allmänna uppfattningen är att öppna program är bättre ur ett säkerhetsperspektiv, men Henriksson (2005) påstår att detta beror på att proprietära produkter oftare blir utsatta för attacker än öppen programvara. Det finns delade åsikter beträffande säkerheten hos öppna respektive proprietära programvaror och vi anser att det är svårt att dra någon generell slutsats vad gäller säkerheten hos en typ av programvara då den är helt beroende av vilken specifik produkt som nyttjas.

En nackdel med öppen programvara är kompatibilitet, som Yngvesson (2005) påpekar i ett exempel med Linux. Han förklarar att ett byte av plattform från Windows till Linux inte skulle fungera eftersom att Linux inte är kompatibelt med högskolans många program, till exempel *LADOK (Lokalt ADB-baserat studieDOKumentationssystem)*. Generellt skulle också ett byte av plattform öka kostnaderna tror Yngvesson (2005).

Den största nackdelen enligt Kihlström & Kihlström (2002) med öppen programvara är bristen på tillförlitlig dokumentation, vilket kan leda till beroende av nyckelpersoner inom verksamheten. Dokumentation kan vara svår att organisera och underhålla då arbetet kan ske distribuerat globalt, vilket kan leda till inaktuell dokumentation. Krav på användning av metoder och modeller, kravspecifikationer eller systemdokumentation vid utvecklandet finns inte, vilket också resulterar i att tillförlitligheten för arbetet minskar. Nyckelpersonsberoende är både positivt och negativt enligt Henriksson (2005) då han menar att de flesta personer i dagsläget är nyckelpersoner på ett eller annat sätt. Det är svårt att förlora nyckelpersoner eftersom de besitter en hel del kunskap, men å andra sidan är det en värdefull tillgång att ha dessa nyckelpersoner inom sin egen organisation.

5.1.6 Utveckling av befintliga/egna öppna program

På läroanstalterna utvecklas egna öppna program i viss mån. Det största projektet är LADOK som är ett "community source" projekt. Med detta menas att det utvecklas, förvaltas och ägs gemensamt av Sveriges högskolor och universitet. U-portal, som är ett portalverktyg, är ett annat "Community source" projekt vilket både Linköping och Uppsala universitet är involverade i. Detta är ett globalt samarbete mellan universitet världen över, vilket innebär utveckling såväl som distribution av systemet. Uppsala har utöver detta projekt ett annat lokalt projekt kallat Bifrost baserat på Linux och annan öppen programvara, vilket är ett försök till att utveckla en ny routingteknik främst för att skapa alternativ till andra aktörer på marknaden. Gemensamt för samtliga lärosäten är att det pågår utvecklingsprojekt i mindre skala, men dessa är då anpassade för specifika syften inom verksamheten. Läroanstalternas initiativ att starta "community source" projekt ser vi mycket positivt på eftersom att detta främjar samarbete mellan läroanstalterna och gynnar den tekniska utvecklingen av

både befintliga och nya öppna program. Det är en bra strategi som möjliggör skräddarsydda system till en lägre kostnad då denna fördelas bland samtliga medverkande.

5.1.7 Licenser för öppen programvara

Samtliga respondenter är osäkra på vilka licenser som nyttjas inom respektive läroanstalt för de öppna programvarorna, men tror att de förmodligen är de mest förekommande licenserna som brukas. Det finns ett stort antal licenser på marknaden idag för öppen programvara och de två som tidigare nämnts, GPL och BSD licenserna, är de mest förekommande (Susning, 2004).

Nejdeby (2005) påstår sig vara en förespråkare av öppen programvara men han tillägger även att det är viktigt att vara realistisk. Med realistisk menar han att om öppen programvara är skyddad av till exempel en GPL-licens, där det finns krav på att omarbetningar måste återdistribueras som öppen programvara, är det omöjligt att kunna tjäna pengar på sitt arbete. Enligt Nejdeby (2005) leder detta till att programvaran inte kan överleva långsiktigt. Nejdeby anser också att öppen programvara projekt ofta är kortlivade om det inte finns en eller flera kommersiella aktörer som backar upp det. Även om det i projektets begynnelse finns en vision om, och strävan efter ära mer än pengar tror Nejdeby (2005) att i längden inser flertalet att de måste tjäna pengar för att överleva. Han räknar upp ett antal projekt som fallerat på dessa grunder, och nämner även Linux som ett exempel på ett projekt som är framgångsrikt än idag. Linux, tillägger han, lever än idag just av den anledningen att det finns vinstdrivande företag bakom projektet. Nejdeby vill tro att fallet skulle vara annorlunda om Linus Torvald inte fått ersättning för sitt arbete. Linux kärnan finansieras bland annat av kommersiella aktörer däribland IBM och Red Hat. Det finns även ett flertal företag vilka säljer sina produkter eller tjänster med anknytning till öppen programvara (Andersson, 2005). Det är omöjligt att bedöma om projektet hade överlevt så pass länge som det har gjort om det inte hade funnits kommersiella företag som backat upp det. Emellertid hade framgången förmodligen inte varit så omfattande utan stödet från dessa företag.

Då diskussioner sker kring begreppen fri programvara och öppen källkod kan begreppen endast särskiljas vad gäller uppfattningar om programvarans egentliga syfte. Enligt teorin kan Nejdeby (2005) anses tillhöra öppen-källkods-rörelsen, vilken har en mer pragmatisk inriktning är fri programvara rörelsen. Anhängare av öppen källkods-rörelsen anser att syftet med denna typ av programvara inte endast är politiskt, utan också kommersiellt. Nejdeby förklarar varför han är skeptisk mot GPL licensen, och i enlighet med den nyttoinriktade öppen källkods-rörelsen anser han att det ska finnas möjligheter för till exempel företag att ta en kollektivt skapad programvara och vidareutveckla denna till en lukrativ produkt utan att själva behöva göra källkoden tillgänglig för andra (Wikipedia, 2005). Bengtsson (2005) å andra sidan tillhör mer den politiska rörelsen eftersom hans intresse för öppen programvara inte främst baseras på kommersiella intressen. Vad gäller Yngvesson (2005) och Henriksson (2005) kan vi inte påstå att dem tillhör någon av ovan nämnda rörelser, utan befinner sig någonstans mitt i mellan beträffande deras åsikter kring öppen programvara.

5.1.8 Attityder gentemot öppen programvara inom organisationen

Både Bengtsson (2005) och Nejdeby (2005) tror att de rådande attityderna inom universiteten är varierande från "ideologiskt engagerade" till dem som är ointresserade av olika skäl. Nejdeby (2005) vill dock påstå att en förändring har skett över tiden vad gäller attityderna inom Linköpings universitet. Han tror att intresset har minskat eftersom bilden av det som tidigare ansågs som tillgängligt, gratis och fritt har förändrats. I dag har många kommit till insikt om att flera kriterier måste tas i beaktning vid utvärdering av och investering i programvara.

Det minskade intresset för öppen programvara som fenomen kan bero på externa kommunikationskanaler. Medias intresse för och användargruppernas utveckling av denna typ av programvara kan ha påverkat hur attityder förändrats över tiden. Sedan media började uppmärksamma fenomenet har information presenterats om dess egentliga koncept och de stora variationer som denna typ av programvara har vad gäller licensformer. Det ökade informationsutbudet kan ha skapat en medvetenhet som tidigare inte existerade beträffande fenomenet, vilket i sin tur har lett till det minskade intresset. Henriksson (2005) tycker sig se ett allmänt intresse för nya produkter och lösningar på IT-området, men han vill inte påstå att det finns någon som genuint brinner för öppen programvara. Det som hellre tas upp i diskussion är de ekonomiska aspekterna av öppen programvara.

Bengtssons (2005) egen uppfattning av attityder hos studenter och professorer är att de generellt är konservativa. Detta visar på att organisationen till stor del består av "efterslämrare" ("late majority" enligt Rogers (1995) modell för olika acceptansgrader vid introduktion av nya tekniska innovationer). Anledningen till att situationen på Uppsala universitet är sådan kan vara att dessa grupper anser att de inte har tid att engagera sig. För att få dem intresserade av öppen programvara måste det kanske införas som obligatoriskt i arbetsschemat.

På internationella Handelshögskolan i Jönköping, som är en fakultet inom Högskolan i Jönköping, anger Yngvesson (2005) att attityderna mot öppen programvara generellt sett inte är negativa men poängterar att en funktionaliteten hos öppen programvara ska vara bra eller bättre för att få positiv respons från de anställda. Här kan paralleller dras till Moore och Benbasat (1991) modell, där relativ fördel ingår som ett av sex begrepp. Yngvesson (2005) menar att innovationen måste uppfattas bättre än tidigare föregångare för att få acceptans, vilket innebär att programvarutypen spelar mindre roll vid investeringsbeslut. Då han inte anser att en av typerna generellt sätt är bättre tas olika beslut från fall till fall beroende på programvarornas relativa fördelar.

5.1.9 Förutsättningar för användandet av öppen programvara

Under 90-talet fanns det ett flertal "öppen källkods projekt". GNOME och KDE är de mest betydelsefulla då de lägger stor vikt vid användbarheten för mindre tekniska människor. Dessa projekt har gjort det möjligt för alla att dra nytta av öppen programvara. Mjukvaran som produceras av dessa projekt avfärdar den vanliga myten om att öppen programvara huvudsakligen fokuserar på server och utvecklingsorienterade system (Gonzalez-Barahona, 2000).

Åsikterna om teknikens mognad i fråga om öppen programvara går isär bland de tillfrågande. Enligt Bengtsson (2005) har öppen programvara idag nått den mognadsgrad som krävs för att använda öppen programvara fullt ut. Mycket positivt har hänt det senaste årtiondet beträffande funktionalitet och användarvänlighet. Han anser att flertalet öppna applikationer som finns idag är likvärdiga med de proprietära alternativen på marknaden. Till skillnad från Bengtsson (2005) anser Henriksson (2005) att applikationsutbudet inte är tillräckligt för att vara ett alternativ till total ersättning av de befintliga proprietära programvaror som idag nyttjas inom högskolan.

Nejdeby (2005) diskuterar förutsättningarna att använda öppen programvara ur ett tekniskt perspektiv. Han anser i enlighet med Bengtsson (2005) att förbättringar av tekniken har skett med åren, men eftersom den typ av programvara fortfarande vanligtvis utvecklas av unga teknikorienterade killar finns det brister i programmen avseende användarvänligheten

Vi anser att tekniken idag har nått en mognadsgrad vilken möjliggör användning av öppen programvara i stor utsträckning inom läroanstalterna. Emellertid vill vi inte påstå att det går att använda denna typ av programvara fullt ut, då det fortfarande finns svårigheter med interoperabilitet och kompatibilitet mellan olika öppna programvaror. Dessutom är det en kostnadsfråga för dessa organisationer då investeringar i öppen programvara, i större utsträckning, snarare är en kostnadsdrivande faktor än tvärtom. Eftersom att investeringar i mjukvara görs med hänsyn till relativa fördelar som funktionalitet och användarvänlighet mer än på basis av anskaffnings- och licenskostnader, kan vi dra slutsatsen att teknikens mognad vad gäller öppen programvara är av stor vikt för samtliga läroanstalter vid investeringsbeslut.

5.2 Utvärdering av mjukvara

Vi bad samtliga intervjuobjekt att värdera de kriterier som enligt vår åsikt är essentiella att beakta vid utvärdering av mjukvara. Detta görs enligt följande rangordningsskala 1 = största vikt, 2 = stor vikt, 3 = mindre vikt, 4 = minst vikt. Den första delen berör endast systemet, medan den andra delen tittar på kriterier av mer organisatorisk art.

5.2.1 Systemet

De kriterier som ska rangordnas är följande; anskaffningspris & licenskostnader, implementering (installation av programvara), förvaltning & drift, support av programvara, säkerhet, prestanda, funktionalitet, kvalitet och stabilitet, kompatibilitet (Interoperabilitet med sluten programvara) och omgivande hårdvara: tekniska förutsättningar för användning (t.ex. kapacitetskrav)

Alla lärosäten är ense om att funktionalitet är av störst intresse att beakta vid utvärdering av programvara. Förutom funktionalitet anser Yngvesson (2005) också att förvaltning och drift av systemet är de viktigaste kriterierna att ta i beaktning vid en utvärdering. Orsaken till varför öppen programvara ibland väljs bort framför proprietära alternativ kan vara att det inte finns någon egentlig standard för användargränssnitt bland öppna programvaruprodukter. Användargränssnittet är avgörande för uppfattningar om systemets användarvänlighet och funktionalitet och relationen mel-

lan kriteriernas prioritet speglas väl i respondenternas rangordningar. Att samtliga tillfrågade anser att systemets funktionalitet är av högst vikt visar på att valet av programvara görs utifrån ett annat perspektiv än ett ideologiskt. Skolorna väljer programvaror efter det behov som finns och den funktionalitet som efterfrågas i verksamheten och inte på basis av inläsningseffekten, vilket är idealbilden enligt Statskontorets rapport 2003.

Henriksson (2005) anger att ”stor vikt” också bör läggas vid systemets kvalitet och stabilitet. Till skillnad från Henriksson (2005) anser Nejdeby (2005) att stor vikt bör läggas vid anskaffningspris & licenskostnader, implementering (installation av programvara), förvaltning & drift och support av programvara. Yngvesson är överens med Nejdeby (2005) beträffande den vikt som bör läggas vid support av programvara, men säkerhet, kompatibilitet och omgivande hårdvara och tekniska förutsättningar för användning är också kriterier som han anser vara av stor vikt.

Mindre vikt bör läggas vid systemets prestanda och kompatibilitet med omgivande hårdvara enligt Henriksson (2005), men han tillägger dock att det inte bör uteslutas eftersom det ofta finns mycket andra applikationer och hårdvara som ska köras samstämmigt. Enligt Statskontorets rapport 2005 nämns att tillförlitligheten och prestandan oftast är högre och bättre hos öppen programvara till skillnad från motsvarande proprietära alternativ.

Drift, förvaltning och säkerhet tillhör även dem de kriterier som bör läggas mindre vikt vid (Henriksson, 2005). I motsats till Henriksson (2005) tycker Nejdeby (2005) att de kriterier av mindre vikt är omgivande hårdvara: tekniska förutsättningar för användning (t.ex. kapacitetskrav). Yngvesson (2005) delar varken Nejdebys eller Henrikssons åsikt då han anser att anskaffningspris och licenskostnader är av mindre betydelse, eftersom dessa endast belastar budgeten en gång.

De kriterier som bör läggas minst vikt vid är supporten av programvara, pris och licensbekostnad samt implementering. Eftersom det eventuella problemet med bristande support av programvara kan lösas genom rätt kompetens på arbetsplatsen och anskaffningspris, licensbekostnad och implementering endast är engångskostnader. (Henriksson, 2005) Yngvesson (2005) instämmer med Henriksson vad gäller värdering av implementering men lägger till ytterligare kriterier som bör läggas minst vikt vid. Dessa är systemets kvalitet, stabilitet och prestanda. I likhet med Yngvesson (2005) tycker Nejdeby (2005) att kvalitet, stabilitet och prestanda är av minst vikt, men utöver dessa lägger han också till säkerhet och kompatibilitet.

5.2.2 Organisation

Följande avsnitt tar upp de kriterier som är av mer organisatorisk karaktär. De kriterier som ska rangordnas är; användbarhet/användarvänlighet (användargränssnitt), utbildning (av t.ex. slutanvändare), kompetens (support och teknisk drift), licenshantering, inläsning (Leverantörsberoende) och beroende av nyckelpersoner.

Alla intervjuobjekt påpekar att det kriteriet som läggs störst vikt vid är att användbarheten/användarvänligheten finns. Henriksson (2005) anser att utbildning bör vara av stor vikt eftersom personalen måste ha kunskap om hur systemet fungerar och

känna sig bekväma med att använda det. Yngvesson (2005) däremot tycker att kompetens och beroende av nyckelpersoner bör rangordnas som ”stor vikt”. Nejdeby (2005) rangordnar i likhet med Yngvesson (2005) kompetens och beroende av nyckelpersoner som stor vikt men lägger till support och teknisk drift utöver dessa. Henriksson (2005) rangordnar kompetens kriteriet längre ner på listan än vad Yngvesson (2005) och Nejdeby (2005) gör, men tycker dock att det inte bör glömmas bort. Ett kriterium som Henriksson (2005) rangordnar som mindre viktigt är beroendet av nyckelpersoner. Yngvesson (2005) tycker att licenshantering anses vara av mindre vikt, medan Nejdeby är av annan uppfattning och rangordnar inläsning som mindre viktigt.

Det kriterium som har minst betydelse enligt både Henriksson (2005) och Nejdeby (2005) är licenshanteringen. Henriksson (2005) anser också att inläsning är av minst vikt att beakta vid utvärdering av mjukvara och Yngvesson (2005) instämmer.

Respondenternas åsikter angående ovanstående kriterier kan sammanfattas enligt följande; kriterier vilka respondenterna generellt sätt anser vara av hög prioritet är funktionalitet, användbarhet/användarvänlighet, beroende av nyckelpersoner, support av programvara, kompetens (tekniskt drift & support) och förvaltning & drift. Licenshantering, inläsning, prestanda och implementering tolkar vi som de kriterier vilka generellt sätt är av mindre vikt för respondenterna.

5.3 IT-investeringar

5.3.1 Riktlinjer för mjukvaruinvesteringar

Vad gäller riktlinjer för investeringar i programvara finns det både formella och informella principer som följs inom varje läroanstalt. På Högskolan i Jönköping finns det inga direkta formella riktlinjer utan det är verksamhetsbehovet som styr investeringar i programvara. På Linköpings universitet styr också verksamhetens behov investeringar i programvara, men informellt gäller att i första hand välja kommersiella produkter för att på så sätt minska kravnivån inom läroanstalten. På Internationella Handelshögskolan (IHH) i Jönköping finns det heller inga direkta formella riktlinjer, men verksamheten ställer krav på så låga totalkostnader som möjligt. På Uppsala universitet är det riktlinjer från den centrala inköpsfunktionen för IT som skall följas vid investeringar i programvara, men om beloppet understiger en viss summa kan detta hanteras friare.

I en IT-investeringsmodell finns det riktlinjer för hur en investering i programvara ska genomföras. En sådan modell har enligt Schniederjans et al. (2004) två brister vad gäller riskhantering. De delar in risk i två klasser vilka är fysisk och operationell risk. Fysisk risk avser sårbarheten hos datorers hårdvara och mjukvara som till exempel sabotage, stöld och piratkopiering. Operationell risk innebär misslyckande i att uppnå till exempel förutspådda fördelar, kostnadsreduceringar och implementation. Dessa risker kan bli hanterbara genom att välja rätt IT-investeringsmodell, men de kan aldrig elimineras helt. De flesta av respondenterna följer ingen specifik IT-

investeringsmodell vilket långsiktigt kan orsaka oförutsedda risker som möjligtvis kunde ha hanterats i och med användandet av en IT-investeringsmodell.

5.3.2 Principer vid upphandlingar av mjukvara

Upphandling av programvara skiljer sig åt vad gäller tillvägagångssättet på respektive högskola/universitet. Henriksson nämner att Högskolan i Jönköping principiellt följer Statskontorets riktlinjer vid upphandling av programvara, men påpekar också att det varierar från fall till fall hur väl riktlinjerna följs.

Vid Uppsala universitet sker upphandling av programvara efter riktlinjer utformade av inköpsfunktionen för IT, denna funktion föreslår utöver kommersiella produkter även öppna alternativ. Linköpings universitet föredrar att göra avrop mot ramavtal istället för att upphandla om programvara. Inlåsnings blir ett stort problem vid upphandling av komplexa IT-system då leverantörsberoende orsakar höga kostnader vid byte av system och anpassning till omgivningens system (Statskontoret, 2003). Vi anser att läroanstalterna bör beakta Statskontorets rekommendationer vid upphandling, eftersom de besitter gedigen kunskap vad gäller problem som kan uppstå vid upphandlingar. Idag får vi uppfattningen om att majoriteten av respondenterna inte följer några direkta riktlinjer för upphandlingsprocessen, vilket kan leda till mindre förmånliga avtal med leverantörer.

5.3.3 Investeringsprocessen: Idé → Beslut → Realisering av beslut

Då vi bad Nejdeby (2005) att beskriva händelseförloppet på Linköpings universitet för mjukvaruinvesteringsprocessen berättade han att den oftast initieras då det finns ett behov i verksamheten. Ett förstudieprojekt genomförs med hjälp av högt uppsatta företrädare från verksamheten. Ledningsprojektet tar fram en kravspecifikation och ett upphandlingsunderlag, därefter är det styrgruppen som tar det slutgiltiga beslutet och kommer med ett tilldelningsbeslut. Sedan följer installationsfasen.

Bengtsson (2005) redogör för händelseförloppet på Uppsala universitet och förklarar att då ett beslut ska utvärderas och realiseras måste det genomgå ett visst antal instanser; Först görs en grundlig utredning, sedan går förslaget på remiss till rektorsrådet. Om det tas ett beslut om att investera går remissen vidare till nästa instans som är rektorn, vilken tar det slutgiltiga beslutet.

På Internationella Handelshögskolan i Jönköping berättar Yngvesson (2005) att önskemål och idéer kring inköp oftast kommer ifrån verksamheten, anställda och/eller studenter, men det är Yngvesson och hans kollega Stefan Nylander som har det yttersta ansvaret och är de slutgiltiga beslutsfattarna vid inköp av mjukvara.

Henriksson (2005) berättar hur processen "från idé till beslut" ser ut och det börjar alltid med att ett behov hos personal eller studenter identifieras. Efter det utvärderas hur behovet ska tillgodoses på bästa sätt, och hur Högskolan i Jönköping ska gå tillväga. Efter inköp sker anpassning och testning av programvara vilket utförs av IT-service. IT-service identifierar också eventuella fel som sedan korrigeras, för att sedan skicka den rättade versionen ut på kontoret och slutligen nå studenterna.

Det finns olika slags beslut som Rogers (1995) kategoriserar som valfria, kollektiva, auktoriserade och beroende innovationsbeslut. I samtliga lärosätens fall används de auktoriserade innovationsbesluten, vilket betyder att det är relativt få som tar det slutgiltiga beslutet för att sedan kommunicera ut detta. De auktoriserade innovationsbeslutsfattandet görs ofta av dem som besitter makten, statusen eller den tekniska expertisen. På Linköpings universitet är det ledningsgruppen som tar det yttersta beslutet, medan det på Uppsala universitet är rektorn som gör detsamma. På Handelshögskolan i Jönköping är det däremot de personer som besitter den tekniska expertisen som gör bedömandet, vilket också sker på Högskolan i Jönköping.

Linköpings universitet, Uppsala universitet, Högskolan i Jönköping och Handelshögskolan i Jönköping är ense om att ett behov hos personal eller studenter först identifieras därefter initieras investeringsprocessen, vilket börjar med en utvärdering om detta behov är något som behöver tillgodoses.

5.3.4 Uppföljning av IT-investeringar

I Statskontorets rapport från 2004 framgår det att det finns en avsaknad av ekonomiska kalkyler beträffande uppföljning av IT-investeringar. Ett tydligt behov av att utarbeta metoder och modeller för framtida mjukvaruinvesteringar har därför identifierats.

Den enda läroanstalten som gör noggranna uppföljningar av IT-investeringars nytta i verksamheten är Linköpings universitet. De använder sig av TietoEnators projektstyrningsmetod vilken innefattar uppföljning av interna projekt. Högskolan i Jönköping och dess fakultet IHH gör inga utförligare uppföljningar, men istället förs diskussioner kring interna IT-projekt under de möten som kontinuerligt hålls.

Som tidigare nämnts är nyttjandet av IT-investeringsmodeller relaterat med risk, det finns alltid en risk att de fördelar som förutspåts och/eller reduceringar av kostnader inte uppnås. Projektet kanske heller inte håller den tidsram som beslutas om vid dess initiering. Slut användarmotstånd och/eller att systemet inte är ett stöd för verksamhetsutvecklingen samt eventuella inkompatibilitetsproblem som senare kan visa sig, är även de risker. Dessa risker finns, men de kan genom att modellen väljs med noggrannhet och omsorg, minimeras.

En IT-modell ska enligt Lundberg (2004) vara ett stöd vid värderingen av affärsnyttan och vi anser att det är av vikt att följa upp investeringar för att både se till den kvalitativa nyttan som till exempel student- och personalnöjdhet och IT-nyttan, vilken beskriver kostnadsförändringen för IT, före och efter en IT-investering. Läroanstalterna bör även värdera kostandsbesparingar och ökade intäkter vid en IT-investering. Om läroanstalterna följde metoder/modeller för IT-investeringar skulle arbetet enligt vår åsikt bli mer effektivt och organisationerna mer attraktiva för både studenter och personal.

5.4 Diffusion

5.4.1 Modell för attitydsindelning av organisation

Rogers (1995) modell för olika acceptansgrader vid introduktion av nya tekniska innovationer delas in i följande kategorier; "innovators", "early adopters", "early majority", "late majority". Samtliga intervjuobjekt ombes att uppskatta hur deras eget lärosäte är uppdelat i procent.

Hur fördelningen ser ut inom universitetet vid en introduktion av en ny teknisk innovation var Nejdeby osäker på, men han trodde att ca: 65% kunde passa in under kategorin "late majority", medan 5% kunde klassas som "innovators". Resterande kategorier "early adopters", "early majority", "laggards", tilldelades 10% vardera.

Linköpings universitets fördelning ligger ganska nära Rogers (1995) normalfördelningskurva. Enligt Rogers (1995) ska de två första grupperna; "innovators" och "early adopters" fokuseras mest på när det gäller att introducera en teknisk innovation och få denna att bli så lyckad som möjligt. Detta eftersom att dessa två grupper har stor inflytande i organisationen beträffande acceptans av nya tekniska innovationer. Vi kan dra slutsatsen att om dessa två grupper är större än normalfallet så går en introduktion av en teknisk innovation smidigare och snabbare att införa. I Linköpings fall sker dock inga större avsteg från den normala fördelningen och därför borde acceptansen vid en introduktion av en teknisk innovation ske inom den typiska tidsramen.

Henriksson (2005) gjorde samma uppskattning på 5% som Nejdeby (2005) vad gäller andelen "innovators". Resterande procentsatser fördelar Henriksson (2005) enligt följande; "early adopters" 15%, "early majority" 25%, "late majority" 50% och "laggards" 5%. På Högskolan i Jönköping är gruppen "laggards" väldigt stark, vilket leder till att de ibland saktar ner processen avsevärt vid acceptans av en ny teknisk innovation. Högskolan i Jönköping har en något högre procentsats än Linköping när det gäller "early adopters" och denna grupp refererar Rogers (1995) till som den kategori som har den största påverkan på "gemene mans" åsikter. Han påpekar också att denna grupp ofta har centrala positioner i organisationen och är en respekterad och inflytelserik grupp. Trots att denna grupp är större har "laggardsgruppen" stort inflytande på Högskolan i Jönköping, och saktar därför ner acceptansen av en ny teknisk innovation inom organisationen. Av detta kan slutsatsen dras att också Högskolan i Jönköping tillhör det normala vad gäller acceptans.

Yngvesson (2005) hade svårigheter att svara på hur fördelningen ser ut inom IHH vid en introduktion av en ny teknisk innovation. Han anger att variationen är stor mellan olika avdelningar. Till exempel är de flesta på informatikavdelningen innovatörer, vilket gör det svårt att beakta organisationen som helhet vid en procentuell indelning. Han anser dock inte att den är extrem åt något håll och tillägger därför att den antagligen är normalfördelad. Rogers (1995) menar att innovatörernas betydelse är viktig och att deras funktion bör framhållas. Av detta kan vi dra slutsatsen att på informatikavdelningen är det troligtvis lite enklare att introducera en ny innovation än i den övriga organisationen. Bengtsson (2005) hade liksom Yngvesson (2005) svårigheter att

göra någon uppskattning, men då han inte kunde se några ytterligheter beträffande mönster av acceptansbeteende, menade han att organisationens procentuella fördelning var normalfördelad.

5.4.2 Kommunikationskanaler

Av samtliga respondenter som deltagit i undersökningen anger tre av fyra att muntliga kommunikationskanaler oftast används och att dessa också anses vara mest effektiva vad gäller spridning av information. Linköpings universitet anger även dem att muntliga kommunikationskanaler är effektiva, men nyttjar främst webben och den interna tidningen som kommunikationskanaler.

Rogers (1995) menar att massmediakanaler ofta är de mest effektiva kanalerna när en stor folkmassa skall informeras, och involverar ett massmedium som television, tidningar eller radio, med flera. Det är också effektivt att använda sig av interpersonella kanaler. Detta får bäst resultat om båda parterna har samma socioekonomiska status, till exempel utbildning. Rogers (1995) säger också att ju mer homogena individerna är desto smidigare accepteras innovationen inom organisationen. Alla lärosäten använder sig av interpersonell kommunikation vilket verkar vara det effektivaste sättet att kommunicera ut beslut. Detta kan bero på att inom ett universitet eller vid en högskola är den socioekonomiska statusen likartad och gruppen homogeniserad, vilket leder till att människor förstår varandra på ett bättre sätt. Linköping använder sig av ytterliga kommunikationskanaler, av vilket vi kan dra slutsatsen att Linköpings universitet har förstått vikten av att kommunicera ut ett beslut via flera olika kommunikationskanaler för att förankra beslutets innehåll i verksamheten

5.4.3 Tid

Det råder inte någon generell tidsbegränsning vad gäller acceptans av nya tekniska innovationer på någon av läroanstalterna. Den årliga omsättningen av studentpopulationen är stor inom organisationer som dessa och av både taktiska och praktiska skäl sker därför lansering av nya tekniska innovationer under sommaren. På så sätt minskar svårigheter med att få acceptans bland studenterna och verksamheten störs inte i samma utsträckning. Om det ändå skulle vara så att innovationen inte accepteras är det enligt Nejdeby (2005) nödvändigt att tänka om, kanske är projektet för teknikstyrt och då är det viktigt att vara lyhörd för verksamhetens behov och inte stirra sig blind på källkod. Enligt Rogers (1995) inkluderas sällan variabeln tid eftersom det ibland kan vara svårt att mäta. Tidsaspekten är dock en viktig del av diffusion eftersom ju snabbare en individ eller organisation får erfarenhet av en innovation desto snabbare kan anpassning och acceptans påbörjas. Eftersom lärosätena generellt introducerar nya innovationer på sommaren underlättar detta för studenterna eftersom de får erfarenhet av den nya innovationen när den nya terminen startar vilket samtidigt leder till att acceptansprocessen påbörjas direkt.

5.4.4 Socialt system

Det sociala systemet innehåller gränser inom vilka diffusion sker. Vilka gränserna är beror bland annat på hur den sociala konstruktionen av systemet ser ut samt hur innovationsbeslutsfattande går till. Dessa ämnen involverar hur förhållanden mellan det sociala systemet och diffusionsprocessen ser ut (Rogers, 1995).

Genom att använda sig av den formella beslutsgången bidrar detta till att ge regularitet och stabilitet i organisationen, men också till att förutse gruppens beteende (Rogers, 1995). I lärosätenas fall kan detta tolkas som ju mer formalisering desto större kontroll har ledningen på personers agerande. På Linköpings universitet finns det både en formell och en informell beslutsprocess vilket resulterar i att vissa beslut tas på det traditionella formella sättet, där en eller ett par bestämmer vad som skall investeras i, för att sedan kommunicera ut det nya beslutet. Den informella beslutsprocessen däremot konsulterar ofta de tekniskt kunniga för att utnyttja deras expertis och följa deras råd. Ibland kan denna informella struktur vara överordnad den formella (Rogers, 1995) men det verkar i Linköpings fall som om den formella strukturen är den dominerade. På Jönköpings Högskola berättar Henriksson om hur den formella beslutsprocessen ser ut, och den kan liknas mycket vid Linköpings situation. Högskolan i Jönköping har en formell beslutsgång som följs vid större beslut, men vid mindre beslut lämnas dessa till en något mindre formaliserad grupp. Enligt Yngvesson (2005) på Handelshögskolan i Jönköping så fattas de flesta beslut på ett formellt sätt. Uppfattningen om de informella besluten är att de tas av honom själv i samråd med någon i IT-styrgruppen, därför kan detta tolkas som om det inte görs några direkta informella beslutsfattande eftersom allt sker i samråd med IT-styrgruppen.

På Uppsala universitet finns det både en formell och informell beslutsprocess. Den formella beslutsprocessen fungerar ungefär som på andra intervjuade lärosäten, där beslut kommer ifrån ledningen som därefter skall följas av personal och studenter. Den informella beslutsprocessen ser dock lite annorlunda ut än på övriga lärosäten eftersom det uppmuntras till att diskutera frågor innan den formella beslutsprocessen startas. Bengtsson (2005) nämner den virtuella IT-fakuleten som exempel på detta och förklarar att den påverkar mer genom informella kontakter, än genom formell beslutsmakt.

På frågan om det är underskattat/överskattat att lägga ned tid på att engagera personalen och få dem att känna sig delaktiga vid förändring, svarade Henriksson (2005) att det är underskattat. Han menar att en organisation alltid bör eftersträva att engagera så många som möjligt för att uppnå bästa resultat. Yngvesson (2005) svarar att han till viss del anser att det är överskattat. Att representanter är involverade anger han vara nödvändigt men att involvera alla det är omöjligt. Om fullständig demokrati råder blir inget gjort. Nejdeby (2005) finner att det är av vikt att lägga ned tid på att engagera personalen och få dem att känna sig delaktiga vid förändring, han medger att detta engagemang är essentiellt för att genomföra förändringar, men att en del beslut ibland måste fattas ändå, utan personalens samtycke. Vilket till viss del överensstämmer med Yngvessons konstaterande ovan.

Ett socialt system består av delar som är interrelaterade till varandra och som tillsammans engagerar sig för att lösa problem och uppnå gemensamma mål. Medlem-

marna i ett socialt system består oftast av individer, informella grupper, organisationer och/eller undersystem. Rogers (1995) påpekar att det ofta är svårt för en individ att acceptera en innovation innan det sociala systemet gjort detsamma. Ett lärosäte kan ses som ett socialt system och det är därför viktigt att få individen att känna sig delaktig i det sociala systemet och därigenom skapa en gemenskap och en slags vänskap. Genom att göra personalen delaktig på detta vis underlättar detta inte bara vid introduktion av en ny innovation utan skapar allmänt en positiv anda inom organisationen. Detta leder till att människor tenderar att göra sitt yttersta för att utföra sitt jobb och hjälpa organisationen framåt.

5.4.5 Modell för acceptans av ny teknologi

Samtliga tillfrågade är eniga om att en teknisk innovations relativa fördelar är det högst prioriterade kriteriet för att en introduktion av en produkt ska ske på ett framgångsrikt sätt. Med relativ fördel menas att innovationen uppfattas bättre än tidigare föregångare (Moore och Benbasat, 1991). Därefter rangordnar Yngvesson (2005) och Nejdeby (2005) systemets image på andra plats. På tredje plats placerar Yngvesson (2005) och Henriksson (2005) kompatibilitet i likhet med Nejdeby (2005) men han rankar även komplexitet på denna nivå. Observabilitet och trialabilitet rangordnar samtliga på nivå fyra. Henriksson (2005) placerar även image på denna nivå, medan Nejdeby (2005) istället placerar komplexitet här. Alla är dock ense om att frivilligheten kommer sist då den inte anses vara av större betydelse för en framgångsrik introduktion.

Moore och Benbasat (1991) utvecklar en konceptualiserad modell vilket är ett verktyg för att studera den initiala diffusionsprocessen inom IT-området. De sex begreppen som den konceptualiserade modellen bygger på är; relativa fördelar, image, kompatibilitet, komplexitet, observabilitet, trialabilitet och frivillighet. Utvärderas dessa begrepp ska de i kombination visa på hur organisationen ser ut vad gäller vilket/vilka begrepp som är viktigast i inledningsstadiet när en teknisk innovation ska bli accepterad av individer såväl som organisationer. Eftersom begreppet "relativa fördelar" rangordnas som det viktigaste begreppet av samtliga lärosäten visar detta på att varken individ eller organisation kommer att acceptera den nya innovationen om den inte uppfattas som bättre än vad som tidigare funnits.

Alla lärosäten rangordnar också observabilitet och trialabilitet på samma nivå, vilket är nivå fyra, detta indikerar att dessa två begrepp ej är av stor vikt och att lärosätena inte är särskilt intresserade av att kunna överblicka resultaten som innovationen medför, och inte heller möjligheten att experimentera med innovationen innan införandet.

Alla lärosäten rangordnar "frivillighet" på samma ställe, det vill säga sist, och detta motiveras med att om de övriga begreppen finns så behövs inte frivilligheten då människor automatiskt kommer att välja den nya innovationen eftersom den presenterar så många fördelar. Trots att lärosätena skiljer sig en aning i sina rangordningar visar detta på att högskolor och universitet i Sverige idag är någorlunda uniforma vad gäller prioriteringsordningen av begreppen som är viktiga i den initiala diffusionsprocessen inom IT-området.

6 Slutsats

I följande kapitel presenteras de slutsatser vi dragit efter att ha analyserat empirin med stöd av uppsatsens teoretiska ramverk.

6.1 Utvecklingen för användandet av öppen programvara historiskt sett inom läroanstalterna

Flera års erfarenhet av öppen programvara finns inom samtliga tillfrågade högskolor och universitet. På Linköpings universitet fick denna typ av programvara inte genomslagskraft förrän på 90-talet, detta trots erfarenhet av öppen programvara sedan universitetet grundades 1975. 90-talets början var en milstolpe i utvecklingen av öppen programvara. Internet och USENET möjliggjorde integration av olika användargrupperns arbeten vilket ledde till att kompletta miljöer kunde byggas ovanpå UNIX med nyttjandet av öppen programvara. Det var också under denna tidsperiod som Linus Torvald utvecklade de första versionerna av Linuxkärnan (Gonzalez-Barahona, 2000). Vi tror att förändringar i utvecklingen av öppen programvara på andra sidan Atlanten kan ha varit orsaken till det ökade intresset för öppen programvara på Linköpings universitet, men även andra läroanstalter inom Sverige, eftersom att intresset ökade i samband med de revolutionära framstegen som gjordes inom öppen programvaruområdet.

De rådande attityderna inom läroanstalterna är varierande från ”ideologiskt engagerade” till dem som är ointresserade av olika skäl. Förändring har dock skett över tiden vad gäller attityderna inom Linköpings universitet. Intresset har minskat eftersom bilden av det som tidigare ansågs som tillgängligt, gratis och fritt har förändrats. I dag har många kommit till insikt om att flera kriterier måste tas i beaktning vid utvärdering av och investering i programvara. Det minskade intresset för öppen programvara som fenomen kan bero på externa kommunikationskanaler. Det ökade informationsutbudet från bland annat media kan ha skapat en medvetenhet som tidigare inte existerade beträffande fenomenet, vilket i sin tur har lett till det minskade intresset.

Åsikterna beträffande teknikens mognad i fråga om öppen programvara går isär bland de tillfrågade. Generellt sett är dock respondenterna ense om att tekniken har kommit långt vad gäller användandet av öppen programvara. Beträffande användning av denna typ av programvara fullt ut inom läroanstalterna finns delade meningar. En del anser att kompatibilitet med andra program är begränsad och att det finns brister i användarvänligheten, medan andra anger att det finns en avsaknad vad gäller applikationsutbudet för öppen programvara. Ett mer omfattande nyttjande av öppen programvara inom läroanstalterna, är enligt vår mening möjlig, eftersom att tekniken idag har nått en mognadsgrad vilken möjliggör detta. Emellertid vill vi inte påstå att det går att använda denna typ av programvara fullt ut, då det fortfarande finns svårigheter med kompatibilitet och interoperabilitet mellan olika öppna programvaror. Förutom detta är det en kostnadsfråga för dessa organisationer, då investeringar i öppen programvara, i större utsträckning, snarare är en kostnadsdrivande faktor än tvärtom. Eftersom att investeringar i mjukvara görs med hänsyn till relativa fördelar som funktionalitet och användarvänlighet mer än på basis av anskaffnings- och li-

censkostnader, kan vi dra slutsatsen att teknikens mognad vad gäller öppen programvara är av stor vikt för samtliga läroanstalter vid investeringsbeslut.

6.2 Orsaker bakom investeringar i öppen programvara inom dessa läroanstalter

Nedan följer de centralaste orsakerna bakom investeringar i öppen programvara i punktform. Dessa presenteras utan inbördes ordning.

- Erbjudna öppna programvaror till studenter
 - Öppen programvara skapar oberoende
 - Källkoden är öppen
 - Öppen programvara är säkerhetsmedveten
 - Nyckelpersonsberoende
 - Teknisk utveckling och samarbete
- Relativa fördelar

6.2.1 Erbjudna öppna programvaror till studenter

Ett skäl till att öppen programvara används på högskolor och universitet i Sverige är i utbildningssyfte. Detta för att få studenterna engagerade i de olika programvarualternativ som finns på marknaden, och därmed ha dessa i åtanke vid framtida situationer. Ett annat skäl till att öppen programvara erbjuds i studiesyfte är att lärosätena ska vara uppdaterade vad gäller den snabba tekniska utvecklingen. Detta ger studenterna möjlighet att införskaffa de mest relevanta kunskaperna på området.

6.2.2 Öppen programvara skapar oberoende

En fördel med öppen programvara är att den skapar oberoende, vilket leder till större frihet eftersom förändringar, uppdatering och uppgraderingar av programvaran kan göras efter eget behov. Ett lärosäte anser att flertalet öppna applikationer som finns idag är likvärda med de proprietära alternativen på marknaden. Detta kan ge ytterligare utrymme att utnyttja friheterna med öppen programvara. Öppen programvara är också en försäkring mot leverantörsbortfall, eftersom det ständigt finns utvecklare som arbetar med att förbättra programvaran.

6.2.3 Källkoden är öppen

Eftersom källkoden är öppen har utvecklare möjligheter till att modifiera programmet och återanvända källkod från tidigare öppen-källkodsprojekt, men också för att kodgranskning sker kostnadsfritt av andra programmerare än de som varit med och utvecklat källkoden. Detta leder i sin tur till att fel både kan hittas och åtgärdas snabbare. Fel i säkerheten hos öppna programvaror brukar dessutom korrigeras mycket snabbare (Kihlström & Kihlström, 2002). Det går också att specialanpassa programvaran helt efter egna behov, detta görs i ganska stor uträkning på högskolor och universitet redan idag. Ett exempel på detta är det så kallade "community source"

projektet, vilket baserar sig på samarbete genom utveckling av koden för att därigenom hitta fel, brister och lösningar gemensamt.

6.2.4 Säkerhetsmedveten

Andra fördelar med öppen programvara är att den är mycket säkerhetsmedveten, och installationer och uppdateringar som tidigare har varit krångligt är mycket enklare utformade i de nyare distributionerna. Vad gäller säkerhetsaspekten av öppen programvara förklarar Kihlström & Kihlström (2002) att säkerheten och kvaliteten höjs på öppna programvaror eftersom källkoden är fri att modifiera. Två av respondenterna håller med om att öppen programvara är mer säkerhetsmedveten, medan de andra två respondenterna generellt sett ej ser avsevärda skillnader i säkerheten.

6.2.5 Nyckelpersonsberoende

Att vara beroende av nyckelpersoner kan inte enbart ses som negativt. Om nyckelpersonerna för vidare kunskap ut i organisationen, kan detta vändas till en fördel. En av respondenterna anser att nyckelpersoner i organisationen är en oerhört värdefull tillgång.

6.2.6 Teknisk utveckling och samarbete

Läroanstalterna är involverade i så kallade ”community source” projekt vilket främjar samarbetet mellan läroanstalterna. Detta samarbete gynnar den tekniska utvecklingen av både befintliga men också nya program. Genom att samla kompetens från olika lärosäten sparas också pengar eftersom kostnaden för utvecklingen kan fördelas på samtliga lärosäten. Det blir också lättare att identifiera olika slags fel eftersom kompetensen är så pass hög bland samtliga involverade, och detta hade inte varit fallet om ett lärosäte själva skulle stå för utvecklingen.

6.2.7 Relativa fördelar

Samtliga lärosäten nämner att en teknisk innovations relativa fördelar är det kriteriet som skall prioriteras högst vid en introduktion av en ny produkt. Med relativ fördel menas att innovationen uppfattas bättre än föregångare. Ett exempel på öppen programvara med relativ fördel är Apache. I Apaches fall har de flesta lärosäten i Sverige bytt ut det tidigare serversystemet för att investera i denna öppna produkt. Apache har visats sig ha fler relativa fördelar än vad ett motsvarande proprietärt alternativ har och har därmed bevisat att öppen programvara inte bara kan vara bra utan även bättre än de proprietära produkter som markanden idag erbjuder.

6.3 Orsaker bakom avsaknaden av öppen programvara inom dessa läroanstalter

Nedan följer de centralaste orsakerna bakom avsaknaden av öppen programvara i punktform. Dessa presenteras utan inbördes ordning.

- Kostnader
- Kompatibilitet
- Kompetens
- Supportfunktion
- Attityder
- Användarvänlighet

6.3.1 Kostnader

I en av statskontorets rapporter (2004) anges att öppen programvara främst nyttjas inom den offentliga förvaltningen för besparingsskäl. Detta påstående går emot det synsätt som de lärosäten vi intervjuat har. Samtliga universitet/högskolor som deltagit i undersökningen anger att investeringar i öppen programvara inte beror på besparingsskäl, tvärtom är denna typ av programvara ett dyrare alternativ för läroanstalterna. En av läroanstalterna hänvisar också till den ökade kravnivå som kan växa fram genom investeringar i öppen programvara, och ur en ekonomisk synvinkel är det därför bättre att använda sig av kommersiella produkter.

På flera lärosäten anges att den största anledningen till att välja bort öppen programvara är de förmånliga avtal som Microsoft erbjuder för deras produkter. Leverantörsberoende orsakar höga kostnader vid byte av komplexa IT-system och anpassning till omgivningens system. Då detta är ett stort problem vid upphandling av mer omfattande IT-system, är inlåsning därför ett skäl att välja bort öppen programvara. Av detta kan vi dra slutsatsen att resultatet av statskontorets rapport ej är applicerbart på de lärosäten som vi intervjuat.

6.3.2 Kompatibilitet

En nackdel med öppen programvara är att den inte alltid är kompatibel med andra programvaror, ett exempel som nämns är LADOK systemet som används på alla lärosäten. LADOK fungerar ej tillsammans med Linux operativsystem. Detta kan vara ett skäl till att inte investera i öppen programvara i större utsträckning. Ett av lärosätena påpekar också att det ibland saknas applikationer som motsvarar de proprietära alternativen vilket kan vara ett ytterligare skäl till att investering av öppen programvara inte sker i den utsträckning som är möjligt

6.3.3 Kompetens

Flera lärosäten nämner att det saknas kompetens inom öppen programvara området. Kihlström & Kihlström (2002) menar att det fortfarande finns svårigheter med att få tag i personal med rätt kunskaper för en del specifika öppna program. Orsaken till detta är att antalet leverantörer som håller i utbildningar vad gäller installation, konfiguration och användning av dessa system, ännu är begränsat. De personer som har kunskap om öppen programvara är en värdefull tillgång för företag och annan verksamhet, och det skapar onekligen ett nyckelpersonsberoende. Det är svårt att förlora nyckelpersoner eftersom de besitter en hel del kunskap. Enligt Kihlström & Kihlström (2002) är en stor brist med öppen programvara bristen på tillförlitlig dokumen-

tation, vilket också ökar beroende av nyckelpersoner inom verksamheten. Vi anser att läroverken bör ha en större roll vad gäller utbildning i öppna program, för att öka kompetensen inom området. Genom att göra detta kanske intresset för denna typ av programvara skulle växa inom både skolväsende och näringsliv. Följaktligen skulle det troligtvis också underlätta för företagen att finna kompetent personal vilket motverkar nyckelpersonsberoende.

6.3.4 Supportfunktion

Något som påpekas av flera läroanstalter är att det saknas en tydlig supportfunktion för öppen programvara. Det är bland annat på grund av detta som öppen programvara kan anses skapa oberoende men också därmed minska tryggheten. Detta är ytterligare ett skäl till varför det inte investeras i öppen programvara i större utsträckning inom läroanstalterna.

6.3.5 Attityder

Attityderna har förändrats över tiden och i dagsläget är kunskapen om öppen programvara större än tidigare tack vare informationssamhället. Ett av lärosätena påstår att intresset har minskat för öppen programvara eftersom helhetsbilden av öppen programvara har blivit tydligare. Tidigare ansågs öppen programvara vara något som var tillgängligt och gratis, men denna uppfattning har ändrats. Insikten om att flera komponenter måste tas i beaktning vid en investering av programvara har påverkat beslutsfattarna. Ett annat lärosäte nämner också att attityder hos studenter och professorer generellt sätt är konservativa. Det finns ett allmänt intresse för nya produkter och lösningar på IT-området, men intresset för öppen programvara är svalt. Ett annat lärosäte menar att funktionaliteten hos öppen programvara ska vara bra eller bättre för att överhuvudtaget bli ett alternativ för användarna. Detta kallas för relativa fördelar i Moore och Benbasats (1991) modell för att studera den initiala diffusionsprocessen inom IT-området. Samtliga lärosäten rangordnar relativa fördelar som det begrepp som är viktigast för att få människor att använda den nya produkten. Detta visar på att varken individ eller organisation kommer att acceptera den nya innovationen om den inte uppfattas som bättre än vad som tidigare funnits. Vilket i sin tur leder till att en IT-investering inte kommer att göras om produkten som väljs inte har en relativ fördel.

6.3.6 Användarvänlighet

Flera av lärosätena anger att öppen programvara inte är speciellt användarvänlig och svårhanterlig för den oerfarne. Detta problem har uppstått på grund av utvecklingarna av öppna program ofta är unga teknikintresserade killar som ej tar i beaktning hur väl användarvänligheten fungerar, utan istället fokuserar mer på vad programmet kan prestera. En annan orsak till att användarvänligheten är begränsad är att det inte finns någon egentlig standard för användargränssnitt bland öppna programvaruprodukter, vilket kan leda till att många organisationer väljer att investera i proprietära alternativ istället. På senare har dock användarvänligheten börjat ses över och några projekt som är under pågående är bland andra GNOME och KDE, vilka uppmärksammar

användbarheten för otekniska människor. Alla lärosäten är ense om att funktionalitet och användarvänligt är det kriteriet som bör tas i beaktning först vid en investering av programvara. Att samtliga tillfrågade anser att systemets funktionalitet är av högst vikt visar på att valet av programvara görs utifrån ett annat perspektiv än ett ideologiskt. Skolorna väljer programvaror efter det behov som finns och den funktionalitet som efterfrågas i verksamheten och inte på basis av inläsningseffekten, vilket är idealbilden enligt Statskontorets rapport 2003.

6.4 Sammanfattande slutsatser

Syftet med uppsatsen har varit att kartlägga orsakerna bakom investeringar i eller avsaknaden av öppen programvara på högskolor och universitet i Sverige. Följande punkter presenteras utan ibördes ordning.

Den mest centrala orsaken till investering i öppen programvara är:

- Att kunna erbjuda öppen programvara i studiesyfte för att lärosätena ska vara uppdaterade vad gäller den snabba tekniska utvecklingen.

De mest centrala orsakerna till avsaknaden av öppen programvara är:

- De förmånliga avtal som Microsoft erbjuder lärosätena för deras produkter.
- Den kostnadsfråga som uppkommer för dessa organisationer, då investeringar i öppen programvara, i större utsträckning, snarare är en kostnadsdrivande faktor än tvärtom.
- Att tekniken idag inte är tillräckligt mogen för användande av öppen programvara fullt ut enligt respondenterna.
- Att val görs på grund av systemets eller produktens relativa fördelar och inte på basis av programvarutyp.
- Otydlig supportfunktion för öppen programvara minskar tryggheten.

De mest intressanta iakttagelser vi gjort utifrån undersökningen och det teoretiska ramverket är att vårt antagande beträffande marknadens mognad för användning av öppen programvara fullt ut inte överensstämmer med respondenternas uppfattningar. De skäl som Statskontoret (2004) anger i sin rapport för investeringar i öppen programvara stämmer inte heller överens med de skäl som läroanstalterna angivit. Samtliga universitet/högskolor som deltagit i undersökningen anger att investeringar i öppen programvara inte beror på besparingsskäl, tvärtom är denna typ av programvara ett dyrare alternativ för läroanstalterna. Skolorna väljer programvaror efter det behov som finns och den funktionalitet som efterfrågas i verksamheten och inte på basis av

Slutsats

inläsningseffekten, vilket Statskontoret (2003) anser vara det bästa metoden vid val av programvara.

7 Avslutande diskussion

I följande kapitel förs diskussion kring gjorda metodval och allmänna reflektioner görs kring uppsatsarbetet. Avslutningsvis ges förslag till vidare studier.

7.1 Metoddiskussion

I det inledande skedet i uppsatsarbetet bestämde vi oss för att avgränsa vårt urval till sex stycken läroverk, varav tre var privata och tre var statliga. På respektive läroverk var avsikten att intervjua en IT-chef och en IT-samordnare för att få en mer nyanserad och rättvis bild av situationen på läroverket. Dessvärre fallerade planerna eftersom endast tre läroverk var anträffbara inom uppsatsens begränsade tidsram. Detta resulterade i att vår tidigare avsikt, som var att även jämföra statliga och privata läroanstalter, inte kunde genomföras då vi ej fick jämn fördelning av studieobjekten. Då vi endast fick möjlighet att träffa en IT-samordnade utöver de tre IT-cheferna kan detta ha påverkat resultatets generaliserbarhet. Efter att ha intervjuat IT-samordnaren och IT-chefen vid en läroanstalt märkte vi att uppfattningarna mellan dem skiljde sig åt i en del avseenden. Detta orsakade svårigheter i analysarbetet eftersom generaliserbarheten minskade. Trots detta erhöll vi information som var intressant för uppsatsen och mycket användbar i analysarbetet.

Användning av den semistandardiserade intervjumetoden gav oss frihet att ställa följdfrågor utifrån den förbestämda frågemallen. Detta anser vi varit till stor fördel eftersom att sakförhållanden och skeenden skulle beskrivas vilket gjorde det essentiellt att formulera följdfrågor specifikt för respektive intervjutillfälle. Trots fördelen har det även orsakat nackdelar under analysarbetet. De specifika förhållanden som existerar vid varje läroanstalt har gjort det svårt att dra generella slutsatser, men samtidigt gett ett djup i information vi erhållit. Vi har försökt att uppnå en rimlighet genom att vi har formulerat frågorna utifrån uppsatsens teoretiska ramverk, detta har gjort att den empiriska studien följt en logisk ordning och varit lättare att koppla och jämföra med teorin.

Frågornas omfattning kan ha resulterat i brister i svar från respondenterna. Då frågorna tog lång tid att besvara märkte vi mot slutet av intervjun att respondenten inte var lika ingående som i början av intervjun. Detta kan ha varit orsaken till att en del av respondenternas svar är motsägelsefulla. För att undvika detta problem kunde vi ha delat upp intervjun så att den utfördes vid två olika tillfällen. Detta hade möjligtvis resulterat i svar med större detaljrikedom. Vi har i efterhand selekterat bort en del frågor då vi märkt att dessa varit överflödiga för uppsatsens undersökningsområde. Vid frågor där respondenterna ombads att återge historiska förhållanden kan informationen variera beroende på subjektiva uppfattningar, detta kan ha påverkat den rimligheten i viss mån. Då sakförhållanden och skeenden beskrivs av respondenter kan det trots gott minne återberättas på olika sätt beroende på den som intervjuas. Trovärdighet har uppnåtts då slumpmässiga fel ej har förekommit i stor utsträckning, eftersom förutsättningarna var likartade vid varje intervjutillfälle. Dessvärre kan en intervju ha påverkat undersökningens trovärdighet då intervjun inte kunde slutföras på grund av tidsbrist. Därför saknas en del svar från Uppsala Universitet.

De sekundärdata som samlats in för det teoretiska ramverket har mestadels bestått av artiklar, avhandlingar, tidskrifter etc. från referenser på Internet, detta är ett medvetet val då det är av vikt för att i uppsatsen ha så uppdaterad och aktuell information som möjligt. Den tekniska utvecklingen av öppen programvara och uppfattningar kring denna förändras ständigt och att använda sig av litteratur från till exempelvis bibliotek skulle därför minska uppsatsens trovärdighet. Beträffande teorier kring IT-investeringar och Diffusionsteorier har dock erkända författare valts ut.

7.2 Egna reflektioner

Vi anser att vi uppfyllt syftet med denna uppsats eftersom vi har presenterat de orsaker som finns bakom investeringar i och avsaknaden av öppen programvara inom de utvalda läroanstalterna. Emellertid skulle vi ha önskat att fler läroanstalter kunde ha deltagit i undersökningen för i större utsträckning kunna göra dra allmänna slutsatser. Även möjligheten till att intervjua både en IT-chef och en IT-samordnare skulle ha bidragit till ett mångfacetterat resultat i högre grad och dessutom ökat uppsatsens trovärdighet. Under arbetets gång upptäckte vi dock att om vi genomfört en mer omfattande undersökning skulle vi inte kunna hålla uppsatsens tidsram. Arbetet med att intervjua, sammanställa och analysera hade med andra ord tagit för lång tid. Trots detta anser vi att den information vi erhållit genom vår intervjuundersökning har varit mycket intressant och högst aktuell. Den semistandardiserade metod vi använt oss av vid genomförandet av intervjuerna gav oss möjlighet till att ställa följdfrågor, vilket bidrog till unik och värdefull information om respektive studieobjekt, för uppsatsen.

Då vi tog kontakt med undersökningens intervjuobjekt blev vi väl bemötta och deras inställning till att medverka var i allmänhet positiv. Det var svårt att i förväg uppskatta hur lång tid respektive intervjutillfälle skulle ta, detta eftersom att den semistandardiserade metoden, som tidigare nämnts, gav möjlighet till att ställa följdfrågor. Detta problem resulterade i att en av undersökningens intervjuer inte kunde slutföras, eftersom att tiden inte räckte till. Det som var mest överraskande beträffande resultatet var att de skäl som Statskontoret angett för anskaffning av denna typ av programvara i sina rapporter, avvek från de orsaker som undersökningens respondenter angett. Vi har lärt oss mycket vad gäller planering, och genomförande av uppsatsarbetet. Noggrann planering är A och O eftersom att det är av vikt att vara ute i tid om undersökningen är tidsbegränsad och för att få urvalet av respondenter att medverka i den empiriska undersökningen. Det är svårt att förutspå de problem som kan stötas på under arbetets gång och därför är det essentiellt att vara ute i god tid för att lättare kunna hantera och bemöta olika svårigheter på vägen.

7.3 Förslag på vidare studier

Kunskapsområdet beträffande öppen programvara är mycket omfattande och denna typ av programvara kan studeras i åtskilliga avseenden. I denna uppsats har undersökningen endast omfattat tre studieobjekt, vilka är Högskolan i Jönköping, Uppsala universitet och Linköpings universitet. Detta har gjort det svårt att göra generaliseringar beträffande synen på öppen programvara och orsaker bakom investeringar i eller avsaknaden av öppen programvara. Förslag på vidare studier är att göra en mer omfattande undersökning där samtliga universitet och högskolor i Sverige undersöks, för att på detta sätt ha möjlighet att i större utsträckning dra generella slutsatser. Då kan även skillnader mellan privata och statliga läroanstalter studeras. Det kunde också vara intressant att studera de olika öppna programvarorna som används inom respektive lärosäte mer ingående, för att senare ha möjlighet att utvärdera dessa enligt utvärderingsmodellen som presenteras i vår uppsats. Detta kan även vara av intresse för dessa institutioner vid framtida investeringar i öppen programvara då det ger vägledning inför val av olika öppna produkter.

Ytterligare område av intresse skulle vara att genomföra undersökningen inom små och medelstora kommersiella företag i Sverige. Till skillnad från läroanstalter får antagligen inte dessa kategorier av företag så pass förmånliga avtal med kommersiella aktörer på mjukvarumarknaden. Detta skulle därför innebära att de till skillnad från läroanstalter skulle kunna reducera IT-kostnader med nyttjandet av denna typ av programvara. Ett annat undersökningsområde för denna typ av programvara som faller utanför den avgränsning vår uppsats har vad gäller geografien är utvecklingen och användandet av öppen programvara inom läroanstalter i utvecklingsländer, både ur ett kommersiellt och politiskt perspektiv. Ett flertal myndigheter i dessa länder uppmuntrar idag till användningen av denna typ av programvara då den innefattar egenskaper vilka ger användaren friheter, till skillnad från proprietära alternativ. Öppen programvara möjliggör också anpassningar av programvaran som är av vikt för dessa länder, beträffande exempelvis språkanpassningar.

Litteraturförteckning

- Andersson, M. (2005) *Det ondskefulla imperiet slipar patentvapnen* Hämtad 2005-05-19 från <http://www.uppsnappat.se/news!print.jsp?newsID=7377>
- Andrén, M. (2003) *MPEG-4-kompatibel settop-box för IP-nät baserad på öppna standarder -en systemstudie* Hämtad 2005-05-17 från <http://www.ep.liu.se/exjobb/isy/2003/3328/exjobb.pdf>
- Aronssons Datateknik (2004) *Red hat Linux*, Hämtad 2005-03-18 från Susnings webbplats:http://susning.nu/susning.fcgi?action=browse&id=Red_Hat_Linux&oldid=Red_Hat
- Broersma, M. (2002) *Eric Raymond: Why Open source will rule*, Hämtad 2005-05-04, från http://news.zdnet.com/2100-3513_22-871366.html
- Christensen L., Andersson. N., Carlsson. C. & Haglund. L. (2001) *Marknadsundersökning*. Lund: Studentlitteratur
- Danielsson, M., (2002) *Open Source Software: Paradigmskifte eller dagslända?* Hämtad 2005-03-11, från <http://www.handels.gu.se/epc/archive/00001985/01/Nr6,MD.pdf>
- DVKV. Datavetenskapliga klubbverket (2003) *Vad är UNIX?*, Hämtad 2005-04-01 från http://www.csd.uu.se/students/klubb/nollning/howto/22_UNIX.shtml
- Easttom, C. (2004). *Moving from Windows to Linux*. Hingham, Mass: Charles River Media
- Gabriel, R. P. & William, N.J. (2005) *Sun Community Source License Principles* Hämtad 2005-05-16 från <http://www.sun.com/software/communitysource/principles.xml>
- Goldkuhl, G (1998) *Kunskapande, Centrum för studier av Människa, Teknik och Organisation* (CMTO) Linköpings universitet.
- Gonzalez-Barahona, J.,M. (2000) *Free Software/Open Source: Information Society Opportunities for Europe? A brief history of open source software*, Hämtad 2005-03-29, från http://eu.conecta.it/paper/brief_history_open_source.html
- Gonzalez-Barahona, J.,M. (2000) *Free Software/Open Source: Information Society Opportunities for Europe? Open source software licences* Hämtad 2005-03-29, från http://eu.conecta.it/paper/Open_source_software_licenc.html

Litteraturförteckning

- Håkansson, S. (1996) *Svenska kommuners IT-strategier*, Hämtad 2005-02-20, från <http://www.kultur.nu/rapporter/slutbetankande/slutbil3.html>
- Högskolan i Jönköping (2005) *Organisationsschema* Hämtad 2005-05-25 från <http://www.hj.se/images/misc/orgschema.gif>
- Kihlström, C., Kihlström, M. (2002) *Öppen källkod och offentlig sektor*, Hämtad 2005-05-04, från <http://www.kihlstrom.com/oppenkallkodochoffentligsektor/OppenKallkodOchOffentligSektor.pdf>
- Kjaer Jensen, M. (1995) *Kvalitativa metoder för samhälls- och beteendevetare*, Lund: Studentlitteratur
- Kjær Jensen, M. (1995). *Kvalitativa metoder, för samhälls- och beteendevetare*.
- Linköpings universitet (2005) *Organisationschema för Linköping universitet* Hämtad 2005-05-25 från <http://www.liu.se/basfakta/organisation/>
- Linux Online, (2005) *Linux Distributions*, Hämtad 2005-04-01 från Linux Online, s webbplats: <http://www.linux.org/dist/index.html> Lund: Studentlitteratur.
- Lundahl U., Skärvad P-H. (1999) *Utredningsmetodik för samhällsvetare och ekonomer* (3. uppl.) Lund : Studentlitteratur
- Lundberg, D.(2004) *IT och affärsnytta : konsten att lyckas med investeringar i IT* Lund : Studentlitteratur
- Lunds universitet, (2003) *Vad är Ladok*, Hämtad 2005-04-22, från <http://www ldc.lu.se/ladok/vadar.htm>
- Microsoft (2004) *Öppna standarder* Hämtad 2005-05-19 från <http://www.microsoft.com/sverige/government/general/opensource.asp>
- Moore.G.C, Benbasat.,I. (1991) *Development of Instrument to Measure the Perceptions of Adopting an Information Technology Innovation*, The Institute of Management Sciences.
- Pagina (2005) *IT-ordlista*. Hämtad 2005-05-25 från <http://pagina.se/itord/default.asp>
- Patel, R. & Tibelius, U. (1987) *Grundbok i forskningsmetodik*, Studentlitteratur: Lund
- phpBB Group (2004) *forum.UNIX.se*, Hämtad 2005-04-01, från <http://forum.UNIX.se/viewtopic.php?t=2559>
- Repstad, P. (1999) *Närhet och distans : kvalitativa metoder i samhällsvetenskap* (3., [rev.] uppl.) Lund : Studentlitteratur

Litteraturförteckning

- Rogers, E.M. (1995) *Diffusion of Innovation*, New York: The Free Press
- Schniederjans M. J., Hamaker J. L., Schniederjans A. M. (2004) *Information technology investment: decision-making methodology* Singapore: World Scientific
- Seigerroth, U. (2005) *Föreläsningsunderlag för kursen Samhällsvetenskaplig kunskapsbildning: Grunder och forskningsmetoder*. Hämtad 2005-03-12, från Internationella Handelshögskolans i Jönköping webbplats: http://jibsnet.hj.se/documents/index.asp?SID=11&GID=kurs_samkuns k&PID=2596
- Statskontoret (2003) *Öppen programvara*, Hämtad 2005-03-02, från <http://www.Statskontoret.se/upload/Publikationer/2003/200308.pdf>
- Statskontoret (2004). *Öppen programvara: Erfarenheter av produkter som bygger på öppen källkod inom förvaltningen* Hämtad 2005-03-15, från <http://www.Statskontoret.se/upload/Publikationer/2004/200421.pdf>
- Statskontoret (2005) *Öppen programvara* Hämtad 2005-03-15, från http://www.Statskontoret.se/Statskontoret/templates/Page___1795.aspx
- Statskontoret (2005) *Öppen standard* Hämtad 2005-05-19 från http://www.Statskontoret.se/Statskontoret/templates/Page___1799.aspx
- Steria AB. (2003) *Vad är öppen programvara?* Hämtad 2005-02-28, från <http://www.steria.se/index.db2?id=1086>
- Susning (2002) *System V*. Hämtad 2005-06-04, från http://susning.nu/System_V
- Susning (2004) *Fri programvara*, Hämtad 2005-04-01, från http://susning.nu/susning.fcgi?action=browse&id=Fri_programvara&ol did=Fri_mjukvara
- Susning (2004) *KDE*. Hämtad 2005-06-04, från <http://susning.nu/KDE>
- Susning (2004) *Programvara*, Hämtad 2005-04-05, från <http://susning.nu/susning.fcgi?action=browse&id=programvara>
- Susning (2004) *StarOffice*. Hämtad 2005-06-04, från <http://susning.nu/StarOffice>
- Susning (2004) *Öppen källkod*, Hämtad 2005-03-03, från http://susning.nu/susning.fcgi?action=browse&id=Fri_programvara&ol did=Fri_mjukvara
- Susning (2005) *AT&T*, Hämtad 2005-06-04, från <http://susning.nu/susning.fcgi?action=browse&id=AT%26T>

- Tempsch's, M. (1998) *Linux 2.0*, Hämtad 2005-03-18, från <http://www.dtek.chalmers.se/~d1temp/Linux1.html>
- Thorell, J. (2005) *MIS*, Hämtad 2005-03-08, från <http://www.pagina.se/itord/default.asp?SokOrd=MIS>
- Toft, P. (2004) *Vad är Linux?*, Hämtad 2005-03-14, från http://www.sslug.dk/artikler/hvad_er_linux
- Uppsala Universitet (2005) *Organisation* Hämtad 2005-05-12, från <http://info.uu.se/fakta.nsf/sidor/organisation.idD5.html>
- Wallén, G. (1996). *Vetenskapsteori och forskningsmetodik*. (2:a uppl.). Lund: Studentlitteratur.
- Wikipedia (2005) *Massachusetts Institute of Technology* Hämtad 2005-06-04, från <http://sv.wikipedia.org/wiki/MIT>
- Wikipedia (2005) *Usenet*. Hämtad 2005-06-04, från <http://sv.wikipedia.org/wiki/Usenet>
- Wikipedia. (2004) *Källkod*. Hämtad 2005-03-17, från <http://sv.wikipedia.org/wiki/K%E4llkod>
- Wikipedia. (2004). *Proprietär*. Hämtad 2005-03-20, från <http://sv.wikipedia.org/wiki/Propriet%E4r>
- Wikipedia. (2005) *Öppen Källkod*. Hämtad 2005-03-17, från <http://sv.wikipedia.org/wiki/K%E4llkod>
- Willén, O. (1991) *Kursbemsida för TDDB53: Vad är UNIX*. Hämtad 2005-03-15, från Institutionen för datavetenskap på Linköpings Universitets webbplats: <http://www.ida.liu.se/~TDDB53/exer/sys-exercises/steg-b3.shtml#UNIX>

Bilaga 1 – Ordförteckning

Nedanstående ordlista består av direktciteringar från Paginas IT-ordlista.

APACHE

"A Patchy Server" är det vanligaste programmet för webbservrar idag (2001). Den har cirka 60 procent av marknaden, vilket är nästan tre gånger så mycket som tvåan Microsoft Internet Information Server. Finns för de flesta plattformar. Apache är skrivet i Linux som är gratis och distribueras enligt principerna för Open Source.

APPLIKATIONER

Tillämpning, ordet applikation används även på svenska. Avser ett program eller något programsystem som hjälper användaren att utföra någonting.

BSD

Berkeley Software Distribution, en verksamhet som startades 1979 vid Berkeley University i Kalifornien för att ta fram en version av UNIX. Det har sedan blivit en kommersiell produkt som har namnet BSD.

FSF

Förkortas FSF, stiftelse i USA som förvaltar programvara inom ramen för Open Source. FSF står bakom GNU och de principer som läggs fast där. Grunden för FSF är att programvaran får spridas fritt och det enda man inte får göra är att förhindra fortsatt spridning. Linux följer principerna från FSF men är inte lika med FSF eller GNU. FSF utvecklar nu en ny kärna som heter HURD för att ersätta Linux i GNU-system.

GNOME

En förkortning för GNU Network Object Model Environment, ett grafiskt användargränssnitt och en uppsättning program för Linux. Avsikten är att göra operativsystemet Linux lättare att använda och liknar i mycket Windows och dess tillbehörsprogram. I GNOME kan man välja ett flertal olika användargränssnitt och de vanligaste hjälpprogrammen. Man kan på skärmen efterlikna Windows 98 eller Mac OS. Dessutom innehåller GNOME samma typer av program som t.ex. Office 97: ordbehandlare, kalkylprogram, databashanterare, presentationsprogram, webbläsare och ett program för e-post.

GNU

En rekursiv (rundgående) deklARATION som betyder GNU is Not UNIX och som skrevs av programmeraren Richard Stallman år 1984. Han hade arbetat i UNIX-världen och ansåg att all programvara och all källkod skulle vara fri. Han och några likasinnade skrev ett UNIX-liknande operativsystem och det utvecklades senare till Open Source (fri källkod). Ett av de mest spridda exemplen på Open Source är Linux. Principen om GNU förvaltas nu av Free Software Foundation (FSF).

GPL

General Public License, den allmänna licens för öppen källkod som gäller för GNU och Open Source.)

IBM

International Business Machine Corporation, ett av världens största företag och det största i databranschen. IBM är genom sin storlek ledande inom datorområdet och startades 1924. På persondatorsidan är huvuddelen av de datorer som säljs IBM-kompatibla genom att IBM med sin PC-familj blivit en industristandard.

KDE

En integrerad datamiljö (användargränssnitt) för UNIX-system (bl.a. Linux) som även innehåller en fönsterhanterare som ger ett Windows-liknande gränssnitt på skärmen. KDE följer Open Source och är därför gratis och sprids bl.a. med distributioner av Linux.

MOZILLA

Den organisation inom ramen för Open Source som sedan 1998 övertagit utvecklingen av webbläsaren Netscape.

PERL

Practical Extraction and Report Language, ett tolkat programmeringsspråk eller skriptspråk som ofta används för att skriva skript till CGI-anpassningar.

UNIX

Ett operativsystem som kan användas för flerprogramkörning, som har ett hierarkiskt uppbyggt filsystem och många användbara hjälpfunktioner. UNIX uppfanns 1961 vid Bell Laboratories och finns i många olika dialekter som inte alltid är helt kompatibla.

X11

Ett fönstersystem (användargränssnitt), kallas även X11, utvecklat vid Massachusetts Institute of Technology, som används i operativsystemet UNIX (och dess dialekter) och som liknar Microsoft Windows. Källkoden för X-Window är öppen, open source, och den är därför mycket spridd. Nästan alla användargränssnitt till UNIX och dess dialekter (som Linux) bygger på X-Window.

Direktcitrat från andra källor;

AT&T

AT&T är ett telefonbolag i USA som vid sidan av sin kärnverksamhet utvecklade bland annat UNIX och Programspråket C och uppföljaren C++ vid Bell Labs (Susning, 2005).

KDE

KDE är en skrivbordsmiljö för fönstersystemet X Window System och fungerar på de flesta Unix-varianter. KDE tillhandahåller ett antal program samt ett grafiskt användargränssnitt. Tanken är att alla KDE-program ska se ut och bete sig på likartade sätt, samt att de olika programmen ska samarbeta med varandra på ett smärtfritt sätt. KDE-projektet startades av Matthias Ettrich 1996 som ett alternativ till den kommersiella skrivbordsmiljön CDE. KDE är fri programvara och kan gratis laddas ner gratis från projektets webbplats (Susning, 2004).

LADOK

Ladok står för Lokalt ADB-baserat studieDOKumentationssystem. Det har en för alla LADOKhöskolor gemensam kärna, bestående av grundstruktur, program och terminalrutiner för studiedokumentationsändamål. Hur stor del av denna kärna man vill och behöver använda, avgör varje högskola själv (så länge de författnings- mässiga kraven är uppfyllda). Till kärnan kan man också lägga egna tillämpningar. Varje högskola har en egen databas, utan förbindelse med andra högskolors studie- dokumentationsregister, d.v.s. det egna registret omfattar i princip endast uppgifter som rör den enskilde studentens studier vid just den högskolan (Lunds universitet, 2005).

MIT

Massachusetts Institute of Technology, MIT, är ett privatägt universitet och forskningsinstitut beläget i Cambridge, Massachusetts, USA. MIT är framstående inom de sex områden de är verksamma inom: Arkitektur, ingenjörskonst, humaniora, konst, samhällsvetenskap och medicin. 57 nobelpristagare kommer från MIT. 70 procent av forskningen vid

MIT finansieras av statliga medel, och 20 procent kommer från privata finansiärer (Wikipedia, 2005).

PROGRAMVARA

Programvara eller mjukvara är benämningen på ett eller flera datorprogram som tillsammans med dokumentation och andra tillbehör utgör en produkt. Handel med och nyttjandet av programvaror regleras dels av bestämmelser om upphovsrätt, dels genom avtal (Susning, 2003).

STAROFFICE

StarOffice är ett kontorsprogram (ett datorprogram) från Sun Microsystems som innehåller bland annat ordbehandlare och kalkylprogram. StarOffice är en kommersiell version av OpenOffice (på samma sätt som Netscape numer är en kommersiell variant av Mozilla). Programmet är dock gratis för skolor och universitet (Susning, 2004).

SYSTEM V

System V är en Unix-variant från AT&T. Det utläses *System five*, eftersom V står för den romerska siffran 5 (Susning, 2002).

USENET

Usenet, Usenet news är ett sätt för grupper av människor att kommunicera med varandra över Internet via e-post-liknande meddelanden. Usenet är ett av de äldsta datorkommunikationssystem som fortfarande används, men har under 1990-talet och framåt marginaliserats av WWW och sändlistor. Usenet innehåller många diskussionsgrupper och nyhetsgrupper i vitt skilda ämnesområden. Det skapades 1979 och består av ett nät av s.k. nyhetsservrar som synkroniserar sig och skickar meddelanden mellan sig. Formatet och överföringen av Usenet påminner mycket om e-post; den stora skillnaden är att Usenet används för kommunikation mellan många olika parter och e-post oftast för kommunikation mellan några få (Wikipedia, 2005).

Bilaga 2 - Modell för utvärdering av öppen och proprietär programvara

		Skillnad öppen och proprietär programvara i fri valsituation	Skillnad öppen och proprietär programvara vid skifte från det ena till det andra
1	Anskaffningspris och/ eller licenskostnader	Mätbar skillnad	Mätbar skillnad
2	Användningsvänlighet och påverkan på indirekta kostnader (t.ex. långa svarstider, obegripliga ikoner)	Kunskap saknas om det finns skillnad mellan öppen och proprietär programvara. Antagandet är att sådana omkostnader snarare är en funktion av design än beroende på typ av programvara	
3	Utbildning av slutanvändare	Kunskap saknas om detta. Antagandet är att detta beror på design snarare än av typ av programvara	Utgift som härrör från en del av inläsningseffekten
4	Krav på utbildning i den interna supportorganisationen eller nya servicekontrakt med leverantörer	Det antages att kompetensuppbyggnaden är mer kostsam för öppen programvara. Generellt är kravet på lokal expertis större för öppen än för proprietär programvara. Kunskap om öppen programvara är mindre än för de vanligaste proprietära produkterna.	Utgift som härrör från en del av inläsningseffekten.
5	Programtekniska förutsättningar		
5a	Omgivande programvarors kompatibilitet, interoperabilitet och säkerställande av detta	Denna kostnad, d.v.s. kompatibilitet mot programvara från tredje part, måste analysera i varje enskilt fall. Generellt kan man säga att öppen programvara i högre grad är baserad på öppna standarder och därför a priori får anses vara mer kompatibel.	
5b	Omgivande hårdvara: Tekniska förutsättningar för användning, särskilda kapacitetskrav och egenskaper och säkerställande av detta	Uppgraderingstrycket medför att proprietär programvara kommer att ställa högre krav på resurser än öppen programvara	
5c	Programvara för underhåll och support	Det finns för närvarande färre valmöjligheter för underhåll och support för öppen programvara	
6	Krav på kompetens för underhåll av programvaran och möjlighet att skaffa detta samt driftskostnader antingen i egen regi eller genom tjänstleverantör	Generellt är kravet på lokal expertis högre för öppen än för proprietär programvara eftersom leverantören ofta tillhandahåller utbildning och assistans. För öppen programvara är kompetensuppbyggnad baserad på det lokala initiativet. Möjligheten att skaffa tjänster från tredje part beror för båda typer av programvaror på om det finns tillräcklig efterfrågan.	
7	Programvarans driftstabilitet, leverantörens felrättningskapacitet och strategi för detta (frekvensen osv.)	Öppen programvara har en hög driftstabilitet. För proprietär programvara är felrättning m.m. avhängigt av leverantörens vilja och kapacitet att modifiera programmet. För öppen programvara är felrättning beroende av om det finns programmerare tillgängliga som kan utföra ändringar.	

Bilaga 3 – Frågeformulär

Universitet/Högskola

Kontaktperson:

Titel:

Universitet/högskola:

Grundades, när och av vem?

Kort historik/bakgrund ev. från informationsbroschyr:

Privat

Statlig

Huvudinriktning/ar (Vetenskapsområde/n):

Årlig omsättning:

IT-budget (monetära resurser avsatta för IT-investeringar):

Uppskattning av mjukvarubudget:

Öppen Programvara

1. Vad är er definition av öppen programvara?
2. Vad har ni för erfarenheter av öppen programvara inom organisationen? (Negativa/positiva aspekter)
3. Använder ni öppna program? (Om nej, gå vidare till fråga 4)
4. Vilken/vilka av följande kategorier av öppen programvara används inom organisationen: Exemplifiera! Vilka öppna program använder ni?

Kontorsprogram	<input type="checkbox"/>	_____
Serveroperativ	<input type="checkbox"/>	_____
Databaser	<input type="checkbox"/>	_____
Portaler	<input type="checkbox"/>	_____
Webbapplikationer	<input type="checkbox"/>	_____

5. Varför använder ni öppna program? (tex.besparingsskäl)
6. Vilka fördelar/nackdelar ser ni med öppna program?
7. Utvecklar ni befintliga öppna program?
8. Utvecklar ni egna öppna program?
9. Vilka licenser använder ni för öppen programvara?
10. Vilka attityder finns inom organisationen gentemot öppen programvara?
11. Har attityderna förändrats över tiden?
12. Anser ni att förutsättningarna för användandet av öppen programvara har förändrats över tiden? Kompatibilitet, säkerhet, användarvänlighet, support, kompetens etc.

Utvärdering av mjukvara

13. De kriterier som bör tas i beaktning för utvärdering av mjukvara enligt Tullgren & Rehnström (2005) är följande;

Vilka kriterier anser ni vara av störst vikt, stor vikt, mindre vikt och minst vikt vid investering i mjukvara, 1-4 där 1 är det kriteriet som ni värderar högst och 4 det som ni anser vara av minst vikt.

- 1= Störst vikt
- 2= Stor vikt
- 3= Mindre vikt
- 4= Minst vikt

System

- Anskaffningspris & licenskostnader
- Implementering (installation av programvara)
- Förvaltning & drift
- Support av programvara
- Säkerhet
- Prestanda
- Funktionalitet
- Kvalitet och stabilitet
- Kompatibilitet (Interoperabilitet med sluten programvara)
- Omgivande hårdvara: tekniska förutsättningar för användning (t.ex. kapacitetskrav)

Vilka kriterier anser ni vara av störst vikt vid investering i mjukvara, 1-6 där 1 är det kriteriet som ni värderar högst och 9 det som ni anser vara av minst vikt osv.

Organisation

- Användbarhet/användarvänlighet (användargränssnitt)
- Utbildning (av t.ex. slutanvändare)
- Kompetens (support och teknisk drift)
- Licenshantering
- Inlåsning (Leverantörsberoende)
- Beroende av nyckelpersoner

IT-investeringar

14. Finns det några riktlinjer och/eller principer som ni följer vid investeringar i programvara? Externa/Interna
15. Om ja, vilken instans har utformat riktlinjerna?
16. Följer ni några principer vid upphandling av programvara?
17. Kan ni beskriva händelseförloppet för investerings processen: **idé** --> **beslut** - --> **realisering av beslut**? (Centraliserad/decentraliserad - konsensus/auktoritetsbeslut)
18. Följs affärsnyttan i verksamheten upp efter att IT-investeringar har genomförts, om ja, hur sker detta? (nyttokalkyler)

Diffusion

Innovation

19. Innovation delas in i fem kategorier enligt Rogers (1995), vilken av följande kategorier tycker ni passar in bäst på er organisation när det gäller att ta till sig nya innovationer?

Var god rangordna och dela in ungefärligt procenttal.

- | | Rangordning | Procent |
|--|-------------|---------|
| • "innovators" (innovatörer) | | |
| • "early adopters" (tidiga accepterande) | | |
| • "early majority" (tidig majoritet) | | |
| • "late majority" (sen majoritet) | | |
| • "laggards" (eftersläntrare) | | |
20. Vilka av de ovanstående kategorier anser ni vara mest betydelsefulla när en ny teknisk innovation ska introduceras och snabbt accepteras inom er organisation?

Kommunikationskanaler

21. Vilka kommunikationskanaler använder ni er av vid introducering av en ny teknisk innovation? (ex; mail, brev, informationsmöte, mun till mun)
22. Vilka kommunikationskanaler anser ni är mest effektiva när en teknisk innovation ska introduceras?
23. Använder ni er av interpersonella kommunikationskanaler?

Tid

24. Hur ser ni på tidsaspekten när det gäller introduktion av en ny innovation, hur lång tid får det generellt ta innan innovationen är accepterad?
25. Sätter ni in speciella åtgärder om innovationen inte accepteras?

Socialt system

26. Hur ser den formella beslutsprocessen ut inför introduktion av teknisk innovation? (centraliserad och decentraliserad)
27. Hur ser den informella beslutsprocessen ut inför introduktion av teknisk innovation? (centraliserad och decentraliserad)
28. Vilka personer har nyckelroller i organisationen vid beslutsprocesser inför introduktion av teknisk innovation? (både informell och formell) (centraliserad och decentraliserad)

29. Är det överskattat/underskattat att lägga ner tid på att engagera personalen och få dem att känna sig delaktiga vid förändring vid införandet av nya tekniska innovationer inom er organisation? Varför?

Modell

30. Moores och Benbasat har utvecklat en av Everett M.Rogers modeller som tar upp kriterier som vägs in när en ny innovation ska introduceras,

Vilka av dessa följande kriterier anser ni vara viktigast för att en teknisk innovation ska accepteras på ett framgångsrikt sätt inom er organisation? (rangordna 1-7, där 1 är av störst vikt och 7 av minskt vikt. Observera att fler kriterier kan placeras på samma nivå)

___ **Relativa fördelar:** I vilken utsträckning innovationen uppfattas vara bättre än föregångare.

___ **Image:** Graden av hur en innovation kan upphöja imagen eller statusen i ett socialt system.

___ **Kompabilitet:** Om innovationen uppfattas vara konsistent med de existerande värderingar, behov, samt tidigare erfarenheter som potentiella kunder haft.

___ **Komplexitet** Vilken svårighetsgrad den nya innovationen uppfattas ha.

___ **Observabilitet:** Hur pass överblickbara resultaten är som innovationen medför.

___ **Trialbilitet:** Hur mycket det går att experimentera med innovationen innan införande.

___ **Frivillighet:** Hur frivillig innovationen känns för användaren.

Bilaga 4 – Empirisk undersökning

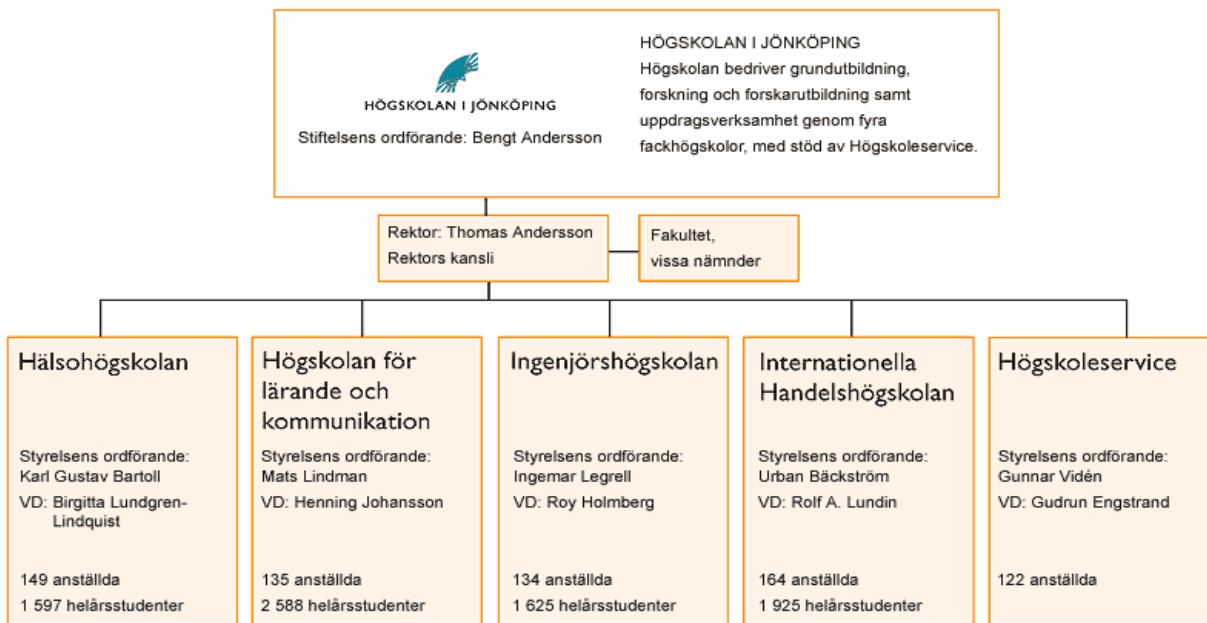
Högskolan i Jönköping

Högskolan i Jönköping grundades 1994 och är en stiftelsehögskola (privat). Skolan kom till under borgerligt styre och var ett ”experiment” som delvis finansieras av staten och delvis av privata investerare. Eftersom Högskolan i Jönköping inte är helägd av staten betyder detta att det finns utrymme för mer frihet än på andra skolor.

Högskolan i Jönköping har cirka 8 000 studenter och cirka 700 anställda. Skolans omsättning uppskattas till 562 miljoner kronor i år. Högskolan erbjuder drygt trettio program och ungefär 300 fristående kurser inom hälsa, vård och socialt arbete, undervisning, medie- och kommunikationsvetenskap, teknik, naturvetenskap och ingenjörsvetenskap samt ekonomi, juridik och informatik

Högskolan i Jönköping består av fem juridiska enheter, fyra fackhögskolorna och högskoleservice. Högskoleservice är högskolans gemensamma serviceorganisation och är en övergripande enhet som tillhandahåller service för de fyra fackhögskolorna. De har hand om och ansvarar för biblioteket, studerandenheten, informationsenheten, lokaler och hyror, drift och underhåll av gemensamma system, ekonomi och personal samt övergripande IT-frågor som berör alla fackhögskolor. Specifika licenser och programvara ansvarar respektive fackhögskola för.

Högskoleservice årliga omsättning är 175 mkr, där drygt hälften går till lokalkostnader. IT-budgeten uppgår till 16 Mkr för högskoleservice och utav den går drygt hälften till personalkostnader. Därefter fördelas budgeten på licenser 1,5 Mkr, gemensam infrastruktur såsom mailsystem, servrar 2-3 Mkr, och resten till övriga investeringar.



Organisationsschema för Högskolan i Jönköping (Högskolan i Jönköping, 2005).

Intervju med IT-chef

Vi har intervjuat Stefan Henriksson, IT-chef på högskoleservice vid Jönköpings högskola den 20 April 2005.

Henrikssons definition av öppen programvara är att den ger tillgång till källkoden, uppmuntrar till modifikation, är kostnadsfri i grunden, har flera ägare, har användarföreningar men ingen förvaltningsorganisation.

Den öppna programvara som idag existerar på Högskolan i Jönköping är väldigt begränsad och används inte i någon större uträkning. Henriksson anger att detta beror på att ett användande i större utsträckning skulle driva upp kostnaderna för högskolan. Idag är de avtal som de har med Microsoft så fördelaktiga att en övergång inte är aktuell av ekonomiska skäl. I driftmiljön används Windows som grundplattform. Den öppna programvara som finns på fackhögskolorna används mestadels i studiesyfte och finns installerat i enstaka datasalar. Henriksson nämner en datasal på Ingenjörshögskolan som använder Linux Operativsystem. Utöver den öppna programvara som finns i datasalar på högskolan används ungefär 10-15 applikationer. De största applikationerna är Apache webbserver, PHP skriptspråk, My-SQL databas, Mozilla Firefox och Novell Netwares tillbehörprogram. I framtiden kommer alla fackhögskolor köra öppen programvara på ett eller annat sätt men aldrig fullt säger Henriksson, han tror också att Linux kommer utvecklas och implementeras i allt större omfattning och det har därför diskuterats att tillsätta en projektgrupp som till hösten ska utvärdera Linux mer i detalj. Detta arbete kommer att ske på Ingenjörshögskolan. Förutsättningarna för att använda öppen programvara har förbättrats över tiden delvis för att det är kompatibelt med Windows och delvis för att det har kommit till fler applikationer, dock ej så många som behövs menar Henriksson. Säkerhetsmässigt är öppen programvara varken bättre än sämre än Microsoft Windows, men Windows råkar generellt ut för fler attacker än öppen programvara. Detta för att Windows är kommersialiserat och har låst källkod.

I dagsläget beräknas TCO (Total cost of ownership) för öppen programvara vara ungefär likvärdigt med proprietära alternativ enligt Henriksson. Han delar in kostnaden i procent och anger att 15-20% läggs på inköp, 40% Drift och resterande 50% på användarstrul.

Vid en eventuell investering av öppen programvara berättar Henriksson att det inte finns några riktlinjer som följs utan att det är verksamhetsbehovet som får styra. Han nämner också att Högskolan i Jönköping principiellt följer Statskontorets riktlinjer vid upphandling av programvara, men påpekar också att det varierar från fall till fall hur väl riktlinjerna följs.

Henriksson berättar hur processen "från idé till beslut" ser ut och det börjar alltid med att ett behov hos personal eller studenter identifieras. Efter det utvärderas hur behovet ska tillgodoses på bästa sätt, och hur Högskolan i Jönköping ska gå tillväga. Anpassning och testning sker därefter vilket utförs av IT-service. IT-service identifierar också eventuella fel som sedan rättas, denna process är iterativ och sker återigen,

för att sedan skicka den rättade versionen ut på kontoret och slutligen nå studenterna. Det görs ingen uppföljning eller nyttokalkyler till dessa investeringsbeslut, men regelbundna möten hålls istället med de fyra olika fackhögskolorna och deras IT-samordnare för att sammanträda om detta. Det finns ett allmänt intresse och nyfikenhet vad gäller nya produkter och lösningar inom IT-området, trots detta intresse finns ganska få som verkligen brinner för öppen programvara som fenomen. Attityder hos anställda på Högskolan i Jönköping är därför inte av intresse att diskuteras i större utsträckning, utan det som diskuteras är mest ekonomiska aspekter hos öppen programvara. Det ska helt enkelt löna sig att införa öppen programvara och därefter diskuteras attityderna till den.

Vi ber Henriksson värdera de kriterier som enligt vår åsikt är essentiella att beakta vid utvärdering av mjukvara. Den första delen berör endast den fysiska delen det vill säga systemet, medan den andra tittar på frågor av mer organisatorisk art.

Det absolut viktigaste kriteriet som Henriksson tar upp är funktionaliteten hos den mjukvara som utvärderas. Systemet ska fungera på alla olika sätt och vis och ska också vara av god kvalitet och ha hög stabilitet säger Henriksson, och påpekar att det är av stor vikt. Vikt bör också läggas vid systemets prestanda och kompatibilitet med omgivande hårdvara. Det kan inte uteslutas eftersom det ofta finns mycket andra applikationer och hårdvara som ska köras samstämmigt. Drift, förvaltning och säkerhet tillhör även dem de kriterier som vikt bör läggas vid. Supporten av programvara är av mindre vikt tycker Henriksson och kan lösas om det finns rätt kompetens på arbetsplatsen. Anskaffningspris och licenskostnad är inte oviktiga kriterier men läggs mindre vikt vid om de övriga punkterna fungerar som de ska. Implementering sker bara en gång och rangordnas därför också som minst viktigt enligt Henriksson.

Det viktigaste är att användbarheten/användarvänligheten finns, annars kommer inte individen använda systemet. Utbildning är väldigt viktigt att ge personalen så att de känner sig bekväma med systemet och får reda på hur det fungerar. Nyckelpersonsberoende är både positivt och negativt, men Henriksson menar att de flesta personer i dagsläget är nyckelpersoner på ett eller annat sätt. Det är svårt att förlora nyckelpersoner eftersom de besitter en hel del kunskap, men å andra sidan är det en värdefull tillgång att ha dessa nyckelpersoner inom sin egen organisation. Den kompetens hos personalen som finns underlättar ofta den tekniska driften och supporten för övriga organisationen. Licenshantering är inget större problem utan det är ofta standardlicenser som används. Inlåsnings ses inte heller som ett stort problem eftersom de flesta leverantörer använder sig av samma metoder. Innovation delas enligt Rogers (1995) in i följande kategorier; *innovators*, *early adopters*, *early majority*, *late majority* och *laggard*s. Henriksson delar procentuellt in Högskolan i Jönköping som följer; Innovators 5%, Early adopters 15%, Early majority 25%, Late majority 50% och Laggards 5%.

Denna rangordning är preliminär eftersom det är väldigt svårt att uppskatta en procentuell fördelning mellan ovanstående alternativ. På Högskolan i Jönköping är gruppen ”laggard” väldigt stark, vilket leder till att de ibland saktar ner processen avsevärt vid acceptans av en ny teknisk innovation.

Henrikssons beskrivning av olika beslutsprocesser på Högskolan i Jönköping ser ut enligt följande; det centraliserade innovationsbeslutsfattandet sker genom att IT-samordnargruppen som leds av IT-chefen bereder frågan för att därefter antingen ta beslut, om beslutet ryms inom budgeten, eller skicka frågan vidare till IT-nämnden, som tar hand om de tyngre ekonomiska beslutet. Det decentraliserade innovationsbeslutsfattande tas ofta av IT-samordnaren på respektive fackhögskolorna. Den informella beslutsprocessen har Henriksson svårt att svara för hur den ser ut men han påpekar att en teknisk innovation på decentraliserad nivå alltid skall stämmas av på IT-samordnarmöte och/eller med IT-service. Detta för att säkerställa att innovationen ej får negativa effekter på högskolans IT-miljö.

De kommunikationskanaler som används är IT-samordningsgruppen där varje IT-samordnare informerar respektive bolag om det som berör dem. Denna grupp har regelbundna möten, vilket också är det effektivaste sättet att kommunicera ut information.

Det absolut viktigaste kriteriet för att kunna introducera en teknisk innovation på ett framgångsrikt sätt är att den har *relativa fördelar*, det vill säga att innovationen uppfattas bättre än tidigare föregångare. Utan detta kriterium är det i stort sett inte någon idé att byta till den nya innovationen eftersom den inte presenterar några fördelar. Efterföljande kriterier är *kompabilitet, komplexitet, observabilitet, image, frivillighet, trialabilitet*.

På frågan om det är underskattat/överskattat att lägga ned tid på att engagera personalen och få dem att känna sig delaktiga vid förändring, svarade Henriksson att det är underskattat. Han menade att en organisation alltid bör eftersträva att engagera så många som möjligt inom den för att uppnå bästa resultat.

Intervju med IT-samordnare

Vi har intervjuat Håkan Yngvesson, IT-samordnare vid Internationella Handelshögskolan (IHH), Högskolan i Jönköping den 21 April 2005.

Yngvessons definition av öppen programvara är att den är gratis, med det menar han att den inte har några initiala investeringskostnader (i alla fall inte om vid nedladdning av källkod och kompilering). Han associerar denna typ av programvara främst med Linux. I intervjun med Yngvesson utgår han ifrån Linux i sin definition av öppen programvara. Shareware är heller inget som Yngvesson anser ska förknippas med öppen programvara. På IHH används öppen programvara i mycket liten utsträckning och Linux finns men används endast i utbildningssyfte inte i driftmiljön.

Avsaknaden av öppen programvara i drift- och utbildningsmiljö beror främst på att kostnaderna skulle öka vid en mer omfattande implementering av denna typ av mjukvara. Med omfattande menas till exempel ett byte av plattform från Windows till Linux. Eftersom att det finns ett stort antal program som troligtvis inte är kompatibla med Linux blir det då nödvändigt att behålla Windows plattformen. Kompetens, underhåll och support av öppen programvara är viktiga kriterier att beakta när det gäller öppen programvara menar Yngvesson, dessa skulle, vid en implementering av

öppen programvara i större utsträckning, också bidra med ökade kostnader för högskolan. Den största anledningen till att skolan väljer bort öppen programvara är de förmånliga avtalen som Microsoft erbjuder. Yngvesson menar att de priser som Microsoft erbjuder är så låga att en övergång till annan leverantör eller annan typ av mjukvara inte är ett alternativ. Microsoft har strategiskt riktat in sig på utbildningsanstalter för att sälja in sina produkter, och trots låga priser är det Microsoft som på lång sikt tjänar mest på avtalet. Yngvesson anser dock att det är essentiellt att regelbundet utvärdera de alternativ som finns på marknaden för att följa med i den snabba tekniska utvecklingen och kunna erbjuda studenter och anställda de bästa alternativen efter skolans förmåga.

Då Yngvesson talar om öppen programvara syftar han främst till Linux och menar att det kommer att dröja innan Linux blir riktigt stort. Detta eftersom att han anser det vara av vikt med strategisk planering, framtidsänkande för utvecklandet av program, att hela tiden ligga steget före (exempel är Oracle och Microsoft). Öppen programvara är endast plagiat från andra programvaror på marknaden, vilket i sin tur leder till eftersläntring. De licenser som används för de öppna programvarorna i verksamheten var Yngvesson inte helt säker på men han antog att det var GNU eller GPL. De attityder som finns gentemot öppen programvara inom organisationen är enligt Yngvesson principiellt sett ej negativa, men den programvara som skall ersätta nuvarande måste inte bara vara lika bra utan bättre än tidigare produkter för att ett byte skall vara aktuellt och få acceptans. Då han får frågan om han tror att ett byte av plattform kommer att ske svarar han bestämt nej, detta på grund av ovan nämnda skäl, men däremot är ett projekt på gång när det gäller Linuxlabbsalar. Yngvesson anser att som utbildningsanstalt är de skyldiga studenterna en variation och bredd när det gäller utbildningsmöjligheter, vilket också har resulterat i detta projekt där en utredning skall genomföras, om en eventuell satsning på öppen programvara, som sjsätts hösten år 2005.

När det gäller IT-investeringar för IHH så finns det inga riktlinjer som måste följas men istället krav från verksamheten om så låga totalkostnader som möjligt. Det finns en avsatt budget för fackhögskolan som de måste hålla sig inom. Vid behov av program så är det viktigt att stämma av med licensavtalet, de flesta program som köps in på IHH är önskemål från dess professorer. Nyttokalkyler existerar inte inom verksamheten enligt Yngvesson beror detta på att verksamheten är för komplex och ett effektivare sätt är de veckomöten samtliga berörda har, dessa är till för att diskutera det mesta rörande IT. Yngvesson och Nylander som också är IT-samordnare vid IHH kan ta beslut angående mindre programvaruinvesteringar på IHH för större investeringar har de en viktig roll som beslutsberedare men han anser ändå att beslutsprocessen är decentraliserad med en central samordning. Önskemål och idéer kring inköp kommer oftast ifrån verksamheten, från anställda och/eller studenter, men det är alltså Yngvesson och Nylander som har den avgörande rollen för inköpen. Investeringprocessen hänger dock ihop med Högskoleservice.

Vi ber Yngvesson värdera de kriterier som enligt vår åsikt är essentiella att beakta vid utvärdering av mjukvara. Den första delen berör endast den fysiska delen det vill säga systemet, medan den andra tittar på frågor av mer organisatorisk art.

Förvaltning och drift av systemet och dess funktionalitet anser Yngvesson vara de viktigaste kriterierna att ta i beaktning vid en utvärdering. Han anger dock att samtliga punkter är nödvändiga att titta på vid en utvärdering av mjukvara. Support av programvara, säkerhet, kompatibilitet och omgivande hårdvara och tekniska förutsättningar för användning är de kriterier som han anser vara av stor vikt. Mindre betydelsefullt är anskaffningspris och licenskostnader eftersom detta endast belastar budgeten en gång. Yngvesson nämner också då implementering endast sker en gång värderas det som minst viktigt likaså systemets kvalitet, stabilitet och prestanda

Användbarhet/användarvänlighet, kompetens och beroende av nyckelpersoner är de punkter som Yngvesson värderas högst vilket betyder att dessa är ytterst viktiga att beakta vid en utvärdering.

Licenshantering och utbildning anses mindre viktigt och minst viktigt är inläsning. Yngvesson påpekar att om systemet är användarvänligt och att nyckelpersonerna har introducerats för systemet på rätt sätt så faller det andra på plats naturligt.

Innovation delas enligt Rogers (1995) in i följande kategorier; *innovators*, *early adopters*, *early majority*, *late majority* och *laggarder*. Organisationen skall i procent delas in efter dessa klasser. Hur fördelningen ser ut inom IHH vid en introduktion av en ny teknisk innovation hade Yngvesson svårigheter att svara på, eftersom att det varierar mycket beroende på den del av verksamheten som studeras. På informatikavdelningen är till exempel de flesta innovatörer, vilket gör det svårt att beakta organisationen som helhet vid en procentuell indelning. Han tillägger att den antagligen är normalfördelad, eftersom att den inte är extrem åt något håll. Konservativa finns men det gör även de innovativa och där mellan är det en blandning.

De kommunikationskanaler som IHH använder sig av är främst lärarna. Dessa informerar om förändringar som berör studenter och är även de som anses vara de mest effektiva informationsspridarna. Informationsmöten och utbildningar hålls även vid nya lanseringar av produkter på skolan. Yngvesson ger ett exempel på introduktionen av Jibsnät (Intranät) och Groupwise (mailprogram) på skolan, här skickades en enkät ut till både personal och studenter innan införandet. Detta menade Yngvesson skulle skapa mindre motstånd vid introduktionen och underlätta acceptansen. Självklart var även enkäten till för att skapa produkter som var anpassade efter användarnas behov och önskemål. Utbildningen var både individuell och gemensam för personal och studenter. Studenter utbildades även för att utbilda exempelvis professorer vid deras arbetsplats, detta eftersom att det fick dem att känna sig lite speciella och därför också blev mindre motstridiga vid implementeringen och användningen av produkterna. Utbildningsprojektet var mycket lyckat, enligt Yngvesson, men inte nödvändigtvis det mest kostnadssnåla sättet.

Enligt Yngvesson på Internationella Handelshögskolan i Jönköping ser den centraliserade innovationsbeslutsprocessen enligt följande; de större frågorna diskuteras i den interna IT-styrgruppen vilket består av IT-samordnare, administrativ chef och grundutbildningsledare. I denna grupp fattas beslut som påverkar ekonomin. Mindre beslut tas direkt av IT-samordnarna. Synpunkter som kommer in från verksamheten vidarebefordras till IT-styrgruppen. På det centraliserade planet är IT-nämnden det ytters-

ta organet vilket består av högskoleservice, VD, IT-chef samt administrativa chefer från respektive bolag. De beslut som tas här skall verkställas på respektive högskola, och rör främst högskolegemensam IT. Det organ som bereder frågan innan den hamnar i IT-nämnden är IT-samordnargruppen vilket består av IT-chefen, nätverksansvarig, helpdeskansvarig samt IT-samordnare för respektive bolag. Den informella beslutsprocessen ser enligt Yngvesson ut som följer; Vissa mindre eller akuta beslut fattas av Yngvesson, IT-samordnare, eller i samråd med någon annan i IT-styrgruppen. Motsvarande informella beslutsordning kan ses för central IT. Ibland kan vissa tekniska innovationer introduceras utan tidigare beslut. Istället för beslut används utvärderingar som ett fåtal individer får gå igenom för att eventuellt sprida vidare den tekniska innovationen till kollegor och studenter.

De kriterier som Yngvesson anser vara viktigast för att få en teknisk innovation att framgångsrikt accepteras inom organisationen är; *relativa fördelar*, det vill säga den utsträckning innovationen uppfattas vara bättre än föregångaren. *Image*, graden av hur en innovation kan upphöja imagen eller statusen i ett socialt system. *Kompatibilitet* och *komplexitet* anser Yngvesson också vara avgörande. Mindre viktigt var *observabilitet* och *triabilitet*. Minst viktig är frivilligheten, den har ingen större roll för en framgångsrik introduktion och acceptans av en ny innovation.

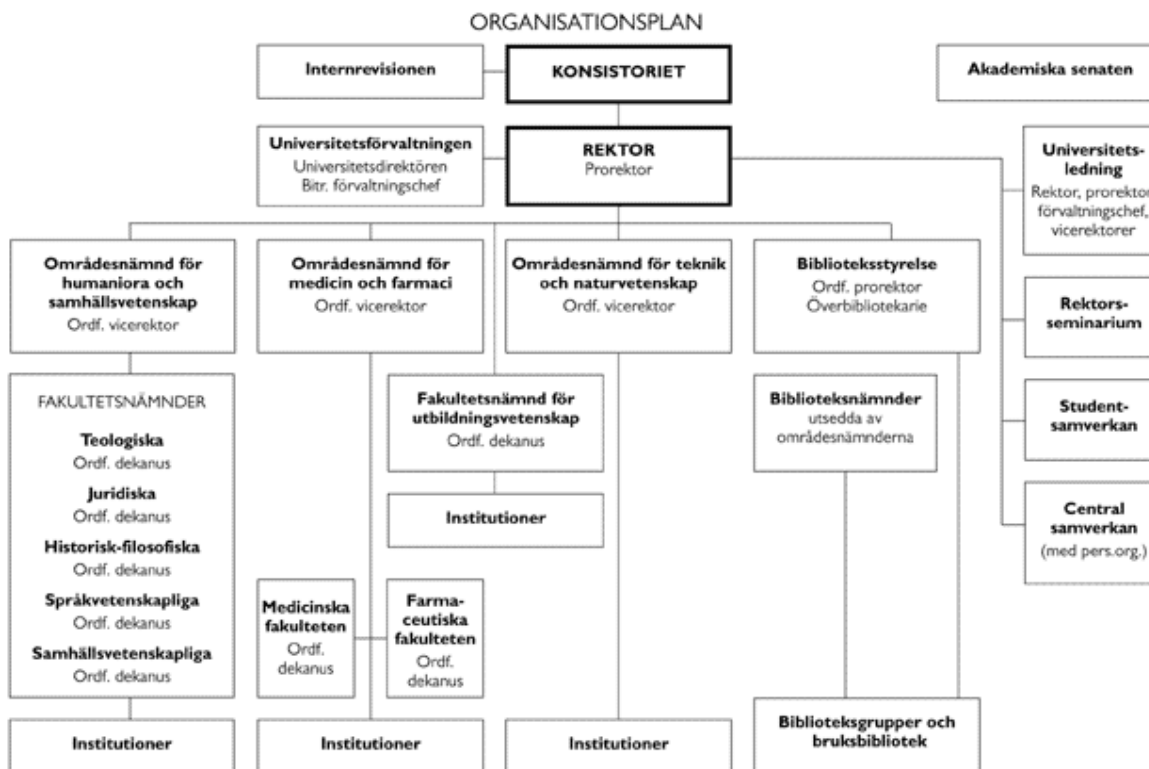
På frågan om det är underskattat/överskattat att lägga ned tid på att engagera personalen och få dem att känna sig delaktiga vid förändring, svarade Yngvesson att han till viss del ansåg att det är överskattat. Att representanter är involverade angav han vara nödvändigt men att involvera alla det är omöjligt. Om fullständig demokrati råder blir inget gjort. Att informera är enligt Yngvesson dock viktigare än att fråga.

Uppsala universitet

Uppsala universitet grundades 1477 och är nordens äldsta universitet. Universitetet är statligt. Uppsala universitet är ett av landets största universitet och har cirka 40 000 studenter och cirka 6000 anställda. Skolans omsättning uppskattas till 4,4 miljarder kronor i år. På Uppsala universitet finns ett stort utbildningsutbud och drygt 40 utbildningsprogram och mer än 1 800 kurser erbjuds.

Utbildning och forskning är uppdelat på nio olika fakulteter, teologiska, juridiska, historisk-filosofiska, språkvetenskapliga, samhällsvetenskapliga, utbildningsvetenskapliga, medicinska, farmaceutiska, teknisk-naturvetenskapliga fakulteterna. Dessa fakulteter är indelade i tre vetenskapsområden vilka är humanistisk samhällsvetenskapliga vetenskapsområdet, medicinska farmaceutiska vetenskapsområdet, teknisk-naturvetenskapliga vetenskapsområdet.

Under universitetsförvaltning i nedanstående organisationsschema finns IT-stöd som har hand om de övergripande IT-frågorna. Denna grupp har också hand om olika sorters investeringar i mjukvara och hårdvara som gäller alla fakulteter. De flesta inköp görs dock direkt av fakulteterna när det gäller en speciell hårdvara eller mjukvara som bara skall användas på fakulteten. IT-stöd står ungefär för 50 miljoner kronor, den totala IT-budgeten är 7% av totala omsättningen, ungefär 280 miljoner kronor.



Organisationsschema för Uppsala universitet (Uppsala universitet, 2005).

Intervju med IT-chef

Vi har intervjuat Ewert Bengtsson, professor i datoriserad bildanalys och rektorsråd för IT-området vid Uppsala universitet den 9 maj 2005. Bengtssons definition av öppen programvara är att det är program där källkoden är fritt tillgänglig, och där vissa licenser finns för att utveckla den fria koden kontinuerligt. Uppsala universitet använder öppen programvara i olika former bland annat i kontorsprogram som Open Office vilket IT-stöd erbjuder gratis till studenter, serveroperativ som Linux vilket används mestadels på den teknisk-naturvetenskapliga fakulteten, databaser som MySQL, portaler som U-portal som är under konstruktion, och en lärandeplattform som Sakai vilket är ett amerikanskt "community source" projekt.

Öppen programvara används inte bara i drift utan också i utbildningssyfte och kurser ges inom den tekniska och naturvetenskapliga fakulteten, dessutom finns det ett antal datasalar med Linux operativsystem på. Sedan flera år tillbaka finns en rekommendation om att använda det öppna filformatet .rtf eller .pdf för dokument snarare Windows filformat.doc. Motivet är bland annat att.doc finns i alltför många versioner vilka är dåligt kompatibla med varandra. Ett annat skäl till att använda .rtf eller .pdf filformat är att .doc är mer virusbenäget och att det dolt i dokumentet sparar gamla

versioner vilket är en säkerhetsrisk. Nyligen har det beslutats att rekommendera att alla formulär som skall fyllas i inom universitetet skall vara utformade så att de fungerar med öppen programvara. Öppen programvara i studiemiljö leder ej till en ekonomisk vinning, men ger på lång sikt människor med en akademisk utbildning vetskapen om att detta alternativ finns. Bengtsson ger också fler exempel på varför öppen programvara i studiesystem är en "god investering för framtiden". Studenterna bör vara medvetna om att leverantörsberoende avstannar utvecklingen och att konkurrens är nyttigt och främjar mångfalden.

Bengtsson säger att målet är att öppen programvara i framtiden ska vara huvuddriftsystemet. Detta underlättar bland annat vid uppdateringar eftersom samtliga system med öppen programvara kan uppdateras genom ett par knapptryckningar istället för en krånglig uppdateringsprocess, miljöerna strular generellt mindre.

Bengtsson påpekar också att det är svårt att skydda sig mot virus när Microsofts produkter används, eftersom de flesta använder det och hackers ofta attackerar dominerande företag. På frågan om Uppsala universitet utvecklar befintliga öppna program svarar Bengtsson att det finns ganska många småprojekt som är väldigt lokalt förankrade. Ett av de lite större projekten är att utveckla portalen U-portal tillsammans med ett antal amerikanska universitet. Utveckling av egna öppna program är till exempel projektet kallat Bifrost som IT-stöd driver, vilket är en router baserad på öppen källkod. Det ses som ett alternativ till Ciscos dominans, Bifrost baseras på Linux och annan öppen programvara. Utvecklingen bygger på en internationell samverkan mellan ca 70 stycken aktiva utvecklare som arbetar "gratis". Denna routingteknik används i flera av universitetets system och därigenom sparas en del pengar. Om det blir billigare eller ej med öppna program råder det delade meningar om inom organisationen. Bengtsson säger att de ekonomiansvariga inte kan se någon vinst i att byta till öppen programvara, snarare en förlust eftersom licenserna för Microsofts produkter redan är så låga och systemet fungerar väl. Bengtsson menar att det är möjligt att den ekonomiska vinsten inte blir så stor men "human kapitalet" ökar i och med att kunskapen ökar hos studenterna.

Nackdelar som finns med öppen programvara är bland annat att kompetensen på området hittills varit ganska så begränsad, detta håller successivt på att förändras. En annan nackdel är det inte finns en tydlig supportfunktion för öppen programvara. Öppen programvara är också svårt att använda för den oerfarne. Fördelar med öppen programvara är att den ofta är mer säkerhetsmedveten, installation har tidigare varit krånglig men i nya distributioner är detta mycket enklare utformat.

Bengtsson anser att marknaden är mogen för att fullt ut använda öppen programvara. Under en 10-årsperiod har funktionalitet och användarvänlighet förbättrats avsevärt vad gäller öppen programvara, och detta håller på att förändra Microsofts dominans på området. Öppen programvara har skapat alternativ till nästan alla olika applikationer som finns i dagsläget, den enda applikation som Bengtsson själv haft vissa kompatibilitetsproblem med är motsvarigheten till PowerPoint, kallt Impress, i OpenOffice.

Bengtsson nämner också att de flesta nya innovationer skapas i öppen programvara, men eftersom kostnaderna för att utveckla dessa innovationer är höga spinner istället större företag, exempelvis Microsoft, vidare på dessa innovationer. Detta kan få de större företagen att framstå som om de var "först" med innovationen. Bengtsson svarar att det beror på okunskap och att det ej är en prioriterad fråga och risk för strul när vi frågar varför öppen programvara inte används fullt ut på läroanstalter.

Vid upphandling av programvara ska riktlinjer från den centrala inköpsfunktionen för IT följas, i vissa fall frångås dessa riktlinjer när investeringen understiger ett visst belopp. Den centrala inköpsfunktionen för IT föreslår också möjligheten att investera i öppen programvara. Uppsala universitet måste också följa de statliga förvaltningslagarna. Bengtsson. Den virtuella IT-fakulteten (se organisationsschema) koordinerar IT-användning, policy och mjuk/hårdvara som används över hela universitet. När det ett beslut ska utvärderas och realiserats måste det genomgå ett visst antal instanser. Först görs en grundlig utredning, sedan går förslaget på remiss till rektorsrådet, om det tas beslut att investera går remissen till nästa instans som är rektorn som tar det slutgiltiga beslutet. Bengtsson.

De attityder som finns inom organisationen gentemot öppen programvara är väldigt varierande och sträcker sig från "ideologiskt engagerade" till ointresserade till totala motståndare säger Bengtsson. Han nämner också att generellt är studenter konservativa och bekväma och gärna använder sådant som de känner igen. Detta gäller även professorer. Bengtsson tillägger dock att det är svårt att generalisera attityderna inom Uppsala universitet därför att attityderna varierar från fakultet till fakultet beroende på dess inriktning. De organisatoriska förberedelser som gjorts för att introducera öppen programvara är till exempel en utbildning i Open Office för anställda. Detta har tyvärr fått dålig genomslagskraft eftersom det är frivilligt och tar upp värdefull tid enligt Bengtsson.

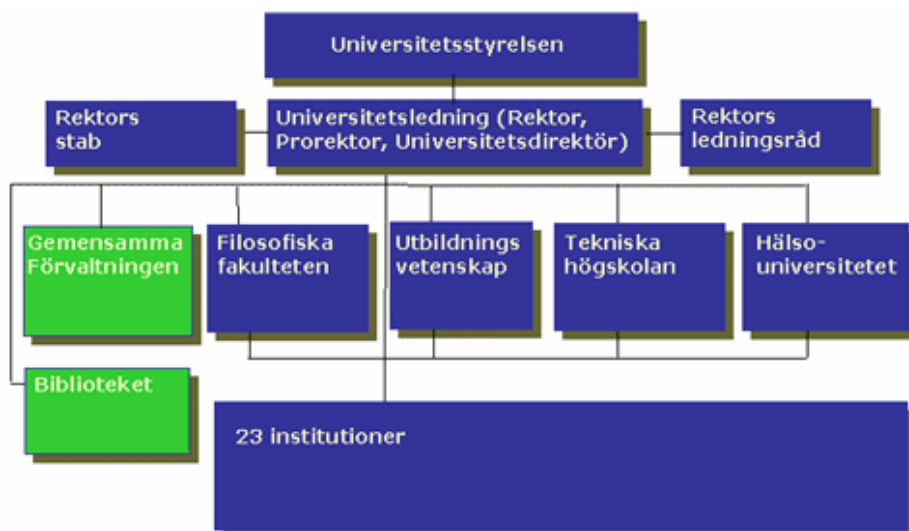
De kommunikationskanaler som används går hierarkiskt och slutar med att professorerna och lärarna informerar studenterna om den nya innovation som kommer att tas i bruk. Innovationsbeslutsfattandet på Uppsala Universitet förklaras av Bengtsson enligt följande; Vad gäller formellt beslutsfattande på central nivå startar denna process alltid med ett behov vilket initieras av verksamheten. Behovet undersöks och bereds för att därefter skriva ett förslag. Detta förslag går ut på remiss vilket leder till ett modifierat förslag, som slutligen skickas till rektorn för formellt beslut. Denna process tar ofta minst ett år att gå igenom. Genomförandet av beslutet beror till största del på hur pass konkret frågan är, det vill säga gäller det ett driftsystem som skall tas i bruk, går detta relativt lätt att införa, men om det gäller riktlinjer så är det svårare att verkställa dessa på en gång. Detta eftersom det tar lång tid att kommunicera ut förslaget, men också eftersom det finns en mängd tolkningar och varianter som måste synkroniseras. Den formella decentraliserade beslutsprocessen fungerar ungefär på samma sätt som den centraliserade, men i detta fall är det beslut i mindre skala som tas, och det är därför lättare att få dessa verkställda. När det gäller informellt beslutsfattande på centraliserad nivå finns det en medveten hållning att idéer skall diskuteras runt i organisationen innan en formell beslutsprocess påbörjas. Detta för att fånga upp relevansen hos frågan och eventuellt upptäcka andra frågor som behövs behandlas. Det finns en instans som kallas för virtuella IT-fakulteten vilket Bengtsson leder,

vilket just har till uppgift att fånga upp dessa frågor och påverka på det informella planet.

Linköpings universitet

Linköpings universitet är statligt och växte fram under 60- och 70-talet. 1970 grundas Linköpings högskola och fem år senare blir högskolan landets sjätte universitet. Linköpings universitet har cirka 26 500 studenter och cirka 3 500 anställda. Skolans omsättning uppskattas till 2.5 miljarder kronor i år (2005). Universitetet erbjuder 100 utbildningsprogram och ungefär 500 fristående kurser. Det finns fyra stycken fakultetshögskolor vilka är; filosofiska fakulteten, hälsouniversitetet, tekniska högskolan och utbildningsvetenskap. Den tekniska högskolan står för ungefär 50 % av studenterna. Linköpings universitet är det tredje största lärosätet i landet inom naturvetenskap och teknik, räknat i antal studenter. Linköpings universitet har sammanlagt 23 institutioner där forskning och utbildning bedrivs.

Den gemensamma IT-budgeten för alla fakultetshögskolor uppgår till cirka 150 miljoner kronor varav 120 miljoner ”delas ut” efter behov. Under universitetsledningen i nedanstående organisationsschema finns förvaltningsledningen som ansvarar för drygt 30 miljoner av IT-budgeten.



Organisationsschema för Linköpings universitet (Linköpings universitet, 2005).

Intervju med IT-chef

Vi har intervjuat Joakim Nejdeby, IT-chef, på Linköpings universitet den 11 maj 2005. Då vi ber Nejdeby att definiera öppen programvara, säger han att han hellre ta-

lar om öppna standarder än öppen programvara. Men han vill klargöra att den allmänna uppfattningen om att denna typ av programvara är gratis inte stämmer.

Proprietära standarder vilket innebär att standarderna inte finns tillgängliga för allmänheten, medför att produkterna endast fungerar tillsammans med andra produkter från samma tillverkare. För att uppnå kompatibilitet mellan produkter från olika tillverkare krävs att de använder samma standard. Med öppna standarder kan vilken tillverkare som helst producera en kompatibel produkt. Det är därför önskvärt med system som baseras på öppna standarder. Med en öppen standard menas en standard som är publikt tillgänglig för vem som helst. Det är viktigt att komma ihåg att bara för att en standard är öppen behöver den inte vara gratis, varken att köpa eller att använda (Andrén, 2003).

Ett annat begrepp Nejdeby tar upp är "Community Source", vilken skapar en gemensam besittning av källkod vitt tillgänglig, precis som öppen programvara modellen gör, men det finns två betydelsefulla skillnader: "Community Source" modellen kräver kompatibilitet bland tidigare utvecklade versioner av programvara och detta upprätthålls genom testning av proprietära modifieringar och vidareutvecklingar inkluderat prestandaförbättringar.

Dessa viktiga skillnader och andra detaljer gör "Community Source" till en mäktig kombination av de bästa proprietära licenserna och modernare än licensieringsmodellerna för öppen programvaruteknologi. Ett exempel på "Community Source" som Nejdeby ger oss är LADOK, vilket är ett gemensamt projekt bland universitet och högskolor i Sverige. Ett annat är uPortal som är ett portalverktyg vilket utvecklas och distribueras av ett stort antal universitet världen över och universitetet i Linköping är en av dem som är delaktiga i projektet (Gabriel & William, 2005).

Nejdeby beskriver sig själv som förespråkare av öppen programvara men han tillägger även att det är viktigt att vara realistisk. Med realistisk menar han att öppen programvara skyddad av T.ex. en GPL licens, där det finns krav på att omarbetningar måste återdistribueras som öppen programvara, gör det omöjligt att kunna tjäna pengar på sitt arbete, vilket enligt Nejdeby leder till att programvaran inte kan överleva långsiktigt. Nejdeby anser att öppen programvara projekt ofta är kortlivade om det inte finns en eller flera kommersiella aktörer som backar upp det. Även om det i projektets begynnelse finns en vision om och strävan efter ära mer än pengar tror Nejdeby att i längden inser flertalet att de måste tjäna pengar för att överleva. Han räknar upp ett flertal projekt som fallerat på dessa grunder, och nämner även Linus Torvald som ett exempel på ett projekt som är framgångsrikt än idag. Linux, tillägger han, lever än idag just av den anledningen att det finns vinstdrivande företag bakom projektet. Nejdeby vill tro att fallet skulle vara annorlunda om Linus Torvald inte fått ersättning för sitt arbete.

Den största nackdelen och fördelen med öppen programvara är oberoende enligt Nejdeby. Oberoende ger dig frihet men minskar tryggheten. Friheten att välja efter det behov som finns i verksamheten och friheten att välja mellan olika produkter. Oberoendet skapar dock en viss otrygghet, om inköp sker av kommersiella företag ingår ofta garantier av olika slag, support etc. Detta försvinner i och med att utveck-

lingen av öppen programvara många gånger inte har något vinstdrivande syfte, vilket leder till att användaren överläts större ansvar.

Linköpings universitet har erfarenhet av öppen programvara sedan universitetets begynnelse men det fick inte genomslagskraft inom organisationen förens på 90-talet vill Nejdeby erinras.

Idag används öppen programvara på universitetet inom de infrastrukturella tjänsterna, exempel är Apache (webbserver) och Linux (Operativsystem), men fortfarande är det Microsofts produkter som dominerar. Öppen programvara används dock i mindre utsträckning för universitetets gemensamma IT-tjänster, men det som används är bland annat, Desktop Management system för Linux, Sun Solaris och Staroffice.

Nejdeby anser inte att licenskostnaderna är av större betydelse för universitet och högskolor eftersom att de avtal som finns men kommersiella aktörer är så pass förmånliga. Inför framtiden tror han att IT-avdelningar är på väg att försvinna, för att istället tas över av SOA (Service Oriented Architecture). Med detta menas att IT-verksamheten kommer att styras efter de tjänster som efterfrågas, där mobilitet blir primärt.

Universitetet utvecklar en del befintliga öppna program som ingår i community source. Även egna öppna programvaror utvecklas och exempel på det är netlogon systemet, vilket Nejdeby anser har en nackdel eftersom det skapar ett beroende av nyckelpersoner. Nätpåloggning (netlogon) är ett system för att autentisera användare av öppna nätanslutningar vid Linköpings universitet och finns tillgängligt under GPL licensen.

Vilka öppna programvarulicenser som används är han osäker på men han skulle gissa på de vanligaste licenserna, sen förekommer nog mindre kända också. GPL licensen ogillar han starkt då den hindrar användaren att tjäna pengar på sin vidareutveckling. Programvaran måste nämligen återdistribueras som öppen programvara. Olika

På frågan om hur attityderna inom organisationen är gentemot öppen programvara, svarar han att flertalet antagligen är ignoranta, men att det finns en hel del "religiösa" är helt klart. Med religiösa avser Nejdeby de som av olika skäl anser att denna typ av programvara är värd mer uppmärksamhet. Han anger att det finns två drivkrafter för utvecklingen av öppen programvara, den ena är pengar och den andra äran. Han tillägger att det oftast inte är fattiga människor som börjar utveckla öppen programvara, utan tvärtom människor med gott om pengar som har "råd" att sträva efter ära. Men som tidigare nämnts, anser Nejdeby att trots att det initialt varit en strävan efter ära, inser dock de flesta programvaruutvecklarna att ära inte räcker för försörjning på länge sikt. Han vill dock påstå att attityderna har förändrats över tiden, idag tror han att den religiösa rörelsen är förbi och att intresset för öppen programvara har minskat i viss utsträckning. Bilden av det som tidigare ansågs som tillgängligt, gratis och fritt har förändrats, eftersom att många kommit till insikt om att flera komponenter måste vägas in än det ovan nämnda vid en utvärdering av programvara.

Förutsättningarna för användandet av öppen programvara har givetvis förändrats över tiden, men utvecklingen av den typen av programvara sker alltså av unga kil-

lar, tekniker. Detta leder till problem då programmen ska lanseras för "vanliga" användare, som kräver användarvänliga program. Många påstår att öppen programvara kan öka konkurrensen på marknaden för mjukvara, då det idag är Microsofts produkter som har monopol på marknaden. Men Nejdeby vill inte se det på det sättet, han ger ett exempel på Apache. Detta program har slagit ut konkurrensen på marknaden för Webbserverar. Idag använder de flesta Apache, vilket har 50% av marknaden för denna typ av programvara.

Vid utvärdering av programvara anser Nejdeby att av de kriterier som vi presenterat rörande systemet, är funktionalitet av största vikt. Anskaffningspris & licenskostnader, implementering (installation av programvara), förvaltning & drift och support av programvara är även dem av stor vikt. Av mindre vikt är omgivande hårdvara: tekniska förutsättningar för användning (t.ex. kapacitetskrav) och av minst vikt är säkerhet, prestanda, kvalitet & stabilitet och kompatibilitet (Interoperabilitet med sluten programvara)

Vad gäller de kriterier av mer organisatorisk art rangordnar Nejdeby användbarhet/användarvänlighet (användargränssnitt) högst. Kompetens (support och teknisk drift) och beroende av nyckelpersoner anser han också vara av stor vikt, därefter kommer utbildning (av t.ex. slutanvändare) och inläsning (Leverantörsberoende). Licenshantering anger Nejdeby som det kriteriet som är av minst vikt

Universitetet har en IT-strategi, men vad gäller riktlinjer för investering i mjukvara råder endast principen om att i första hand köpa in kommersiell mjukvara. Denna princip grundar sig på det faktum att kommersiell programvara har en organisation bakom sig vilket ger ett långsiktigt stöd för användaren av denna. En annan orsak bakom denna princip är att minska kravnivån inom organisationen, då en hög sådan driver upp kostnader De krav och önskemål vad gäller inköp av programvara som kommer ifrån verksamheten, kan begränsas vid inköp av huvudsakligen proprietär programvara. Med kommersiell programvara utblir möjligheter att skraddarsy programvara efter de olika behov som finns inom verksamheten, vilket också resulterar i att olika programvaruprojekt får gå samman och enas om de krav som är av störst vikt. Den instans som utformat principerna bakom inköp av programvara är skolans tidigare vice rektor för IT-frågor. På frågan om några principer följs vid upphandling av programvara svarar Nejdeby att avrop sker gentemot ramavtal men upphandlingar försöker undvikas. Verksamhetens krav styr dock val av produkt, därefter beaktas givetvis även produktens funktionalitet.

Då vi bad Nejdeby att beskriva händelseförloppet för mjukvaruinvesteringsprocessen berättade han att den oftast initieras då det finns ett behov, vilket för det mesta är baserat på trender, som t.ex. i fallet med elektronisk fakturahantering. Ett förstudieprojekt genomförs med hjälp av högt uppsatta företrädare från verksamheten. Ledningsprojektet tar fram en kravspecifikation och ett upphandlingsunderlag, därefter är det styrgruppen som tar det slutgiltiga beslutet och kommer med ett tilldelningsbeslut. Sedan följer installationsfasen.

För att följa upp nyttan vid investeringar har universitetet anlitat Tieto Enator. Ledningen för administration och förvaltning vid Linköpings Universitet har valt Tieto-

Enators metod PPS (Praktisk ProjektStyrning) som projektstyrningsmetod. Universitet får genom avtalet tillgång till en väl beprövad metod för planering, genomförande och uppföljning av interna projekt.

Nejdeby anser att det är av vikt att följa upp affärsnyttan i verksamheten efter investeringar har gjorts och att detta hanteras av en extern organisation är att föredra, då Nejdeby anser att universitetets egna forskare inte ska vara inblandade i skolans interna verksamheter i den utsträckningen.

Vi bad Nejdeby att indela organisationen enligt Rogers (1995) modell där innovation delas in i följande kategorier; innovators, early adopters, early majority, late majority och laggards. Organisationen skall i procent delas in efter dessa klasser. Hur fördelningen ser ut inom universitetet vid en introduktion av en ny teknisk innovation var Nejdeby osäker på. Men han trodde att ca: 65 procent nog kunde passa in under kategorin "late majority", medan 5 procent kunde klassas som "innovators". Resterande kategorier "early adopters", "early majority", "laggards", tilldelades tio procent vardera.

Vid introduktion av nya tekniska innovationer används främst webben och den interna tidningen som kommunikationskanaler. E-post anser han vara en mindre effektiv kommunikationskanal, då antalet studenter på skolan uppgår till ca: 30 000 och om informationsbrevet besvaras av endast en del av denna population skulle det leda till att webbservern kraschar. Ett annat sätt som Nejdeby tycker är effektivt är muntliga kommunikationskanaler, som att prata med folk, chefer talar med underchefer och de i sin tur med sina anställda.

Det råder inte någon generell tidsbegränsning vad gäller acceptans av nya tekniska innovationer. Åttio procent av studentpopulationen byts ut varje år och av taktiska skäl sker därför lansering av nya tekniska innovationer under sommaren, på så sätt minskar svårigheter med att få acceptans bland studenterna. Om det ändå skulle vara så att innovationen inte accepteras är det enligt Nejdeby nödvändigt att tänka om, kanske är projektet för teknikstyrt och då är det viktigt att vara lyhörd för verksamhetens behov och inte stirra sig blind på källkod.

Den formella beslutsprocessen inför en introduktion av en teknisk innovation är till viss del centraliserad på universitetet. Det finns två beslutstoppnoder bestående av IT-chefen och universitetsdirektören. Det finns även en systemägarstyrelse som omfattar de ovan nämnda positionerna samt representanter från respektive fakultet. Om styrelsen skulle ha svårigheter att komma överens om ett enigt beslut är det universitetsdirektören som får ta ett beslut som sedan vidarebefordras till universitetets rektor vilken tar det avgörande beslutet.

Nejdeby anger att det självfallet finns en informell beslutsprocess inför introduktion av en ny produkt. Vidare säger han att vid investeringar så tillfrågas de med mest kunskap på området, så ur detta perspektiv är den informella beslutsprocessen mer decentraliserad än den formella, vilket många gånger är fallet på de flesta företag.

De kriterier som Nejdeby anser vara viktigast för att få en teknisk innovation att framgångsrikt accepteras inom organisationen är; relativa fördelar, det vill säga den

utsträckning innovationen uppfattas vara bättre än föregångaren. Image, graden av hur en innovation kan upphöja imagen eller statusen i ett socialt system. Kompatibilitet bedömer Nejdeby också vara avgörande. Minde viktigt var komplexitet, observabilitet och triabilitet. Minst viktig är frivilligheten, den har ingen större roll för en framgångsrik introduktion och acceptans av en ny innovation.

Nejdeby finner att det är av vikt att lägga ned tid på att engagera personalen och få dem att känna sig delaktiga vid förändring, han medger att detta engagemang är essentiellt för att genomföra förändringar men tillägger att en del beslut ibland måste fattas ändå, utan personalens samtycke. Han ger ett exempel på nedskärningar inom organisationen, i detta fall är det omöjligt att få de anställda att vara positiva eller i alla fall engagerade i förändringen.

Bilaga 5 – Tabellsammanställning för utvärdering av mjukvara

Systemet

- 1= Störst vikt
2= Stor vikt
3= Mindre vikt
4= Minst vikt

	Joakim Nejdeby	Håkan Yngvesson	Stefan Henriksson
Anskaffningspris & licenskostnader	2	3	4
Implementering	2	4	4
Förvaltning & Drift	2	1	3
Support av programvara	2	2	3
Säkerhet	4	2	3
Prestanda	4	4	3
Kvalitet och Stabilitet	4	4	2
Funktionalitet	1	1	1
Kompatibilitet	4	2	3
Omgivande hårdvara	3	2	3

Organisation

- 1= Störst vikt
2= Stor vikt
3= Mindre vikt
4= Minst vikt

	Joakim Nejdeby	Håkan Yngvesson	Stefan Henriksson
Användbarhet/användarvänlighet	1	1	1
Utbildning	3	3	2
Kompetens (tekniskt drift & support)	2	2	3
Licenshantering	4	3	4
Inläsning	3	4	4
Beroende av nyckelpersoner	2	2	4

