

Användning och utveckling av öppen programvara

Mathias Bonn

Kandidatavhandling i datateknik

Handledare: Annamari Soini

Fakulteten för naturvetenskaper och teknik

Åbo Akademi

Innehållsförteckning

1. Inledning.....	2
2. Bakgrund.....	3
2.1. Koncept.....	3
2.2. Exempel.....	4
2.3. Avgränsning.....	5
3. Användare.....	5
3.1. Introduktion.....	5
3.2. Kostnad och företag.....	6
3.3. Hårdvara.....	7
3.4. Användarupplevelse.....	10
3.5. Transparens och ideologi.....	11
4. Utvecklare.....	12
4.1. Introduktion.....	12
4.2. Kunskap, erfarenhet och personliga behov.....	13
4.3. Gemenskap, nöje och ideologi.....	13
4.4. Tidsbrist.....	14
4.5. Inkomst och företag.....	14
5. Jämförelse och diskussion.....	14

1. Inledning

Öppen programvara har existerat i decennier, med varierande framgång på användarmarknaden. Många applikationer som utvecklas idag utnyttjar öppen programvara, men förblir slutna i helhet. Fullständigt öppna alternativ existerar i flera fall, men varför används dessa inte oftare? Varför finns det människor som föredrar öppen programvara? Varför utvecklar man öppen programvara i allmänhet?

Syftet med denna avhandling är att undersöka öppen programvara ur två vinklar, användning och utveckling. Målet är att identifiera motivationer samt motvilja till användandet och utvecklandet av öppen programvara på personlig nivå.

Avhandlingen utmynnar i en diskussion, där likheter och olikheter hos motivationer mellan användare och utvecklare identifieras.

2. Bakgrund

2.1. Koncept

Programvara innehåller instruktioner som en dator använder för att utföra en uppgift. Några exempel på programvara i vardagliga sammanhang inkluderar webbläsare, videospelare och dokumentredigerare. Programvara distribueras i huvudsak i binärt format som maskiner förstår, men inte användaren. För att enkelt modifiera programvaran krävs källkoden, instruktioner för hur programmet ska fungera i klartext. Dessutom krävs rättigheter för att göra ändringar i programvaran och distribuera den.

Sluten programvara medför restriktioner på hur användaren kan utnyttja, modifiera och dela programvaran. I många fall publiceras inte källkoden, utan den förblir i privat förvar hos utvecklaren. I enstaka fall kan källkoden publiceras, med olika begränsningar för hur koden kan användas. Till skillnad från sluten programvara, bygger öppen programvara på transparens. En nödvändig förutsättning är att källkoden är tillgänglig för allmänheten, samt att intresserade parter fritt kan studera koden, förstå hur den fungerar, och göra ändringar i den. Programvaran kan delas och användas utan förhinder.[1]

En viktig komponent som krävs för att programvara kan anses vara öppen är en programlicens, som anger användarens rättigheter och skyldigheter. Öppna programlicenser innehåller grundläggande rättigheter som tillåter användaren att använda programmet i vilket syfte som helst, modifiera källkoden och dela ändringarna.[1] Exempel på licenser som används i stor utsträckning är GPL,

MIT och Apache. Därmed är det viktigt att poängtera att programvaran inte nödvändigtvis är öppen även om källkoden är tillgänglig.

Skyldigheter varierar mellan olika licenser. Exempelvis är GPL en licens som kräver att källkoden publiceras då exekverbara filer baserade på källkoden publiceras, medan MIT inte kräver detta.[2]

(TODO: förklara kort skillnad mellan dessa licenser)

2.2. Exempel

(TODO: inkludera exakt statistik)

Ett exempel på öppen programvara som används i större utsträckning är kärnan Linux, som används i flera operativsystem. På persondatorer har Linux-baserade system fortsättningsvis en låg procentandel användare jämfört med konkurrerande operativsystem, såsom Windows och macOS, medan andelen är större för servrar.[3] Linux är ett exempel på hur öppen programvara kan modifieras och utnyttjas i flera andra projekt, såsom mobiloperativsystemet Android och olika Linux-distributioner, exempelvis Ubuntu och Fedora.

Linux skapades och utvecklades ursprungligen självständigt av Linus Torvalds, men antalet utvecklare har ökat sedan dess. Till skillnad från mindre projekt, som vanligtvis utvecklas på fritiden utan större inkomster, utgörs en del av utvecklingen av anställda på större företag, såsom Intel och Red Hat.[4]

Ett exempel på öppen programvara på applikationsnivå är GIMP, ett bildredigeringsprogram. Till skillnad från Linux har GIMP endast en handfull utvecklare, som arbetar på programvaran på fritiden.[5] Projekt som dessa kan

ha svårigheter att konkurrera med slutna programvara, såsom Photoshop, som utvecklas av större team på arbetstid.[6]

Ytterligare ett exempel på öppen programvara på applikationsnivå är LibreOffice, förgrenat ur projektet OpenOffice.org år 2010.[7] LibreOffice erbjuder alternativ till olika kontorsapplikationer i Microsoft Office, till exempel Word, Excel och PowerPoint. LibreOffice fokuserar huvudsakligen på öppna filformat, som OpenDocument Format (ODF), men stöder delvis motsvarande filformat i Microsoft Office, såsom DOCX. Dock är kompatibiliteten med DOCX och övriga slutna format inte perfekt, och kräver regelbunden underhållning för att undvika att halka efter förändringar i formaten.[8]

2.3. Avgränsning

I avhandlingen ligger fokus på två huvudgrupper, användare och utvecklare av öppen programvara. Varför använder man denna programvara? Vilka är anledningarna till att inte använda öppen programvara? På motsvarande sätt tas utvecklare upp. Vilka anledningar finns det att utveckla öppen programvara, och varför är det inte möjligt i alla fall?

3. Användare

3.1. Introduktion

Öppen programvara används av flera olika sorters användare. Tre användargrupper kan definieras: användare som utnyttjar extern källkod i egna projekt, användare som har enkla krav på funktionalitet, samt expertanvändare som har större krav på funktionalitet och exempelvis använder programvaran i

sitt yrke. Eftersom utvecklare behandlas i ett senare skede, kommer den förstnämnda gruppen inte att behandlas i detta avsnitt.

Avsnittet är uppdelat i huvudteman, som jämför anledningar till att vanliga användare och expertanvändare använder/inte använder öppen programvara.

3.2. Kostnad och företag

I allmänhet är öppen programvara kostnadsfri för användaren. Detta beror främst på användarens grundrättigheter, vilka inkluderar rätten att ändra och dela programvaran. För en del användare är kostnad en anledning att använda öppen programvara, om man inte vill spendera en månadsavgift för exempelvis Photoshop, eller olovligt använda betald programvara.[9]

Expertanvändare och vanliga användare har olika krav på funktionaliteten i programvaran. Dessa kostnadsfria alternativ kan vara tillräckliga i vissa fall, särskilt om man stundvis behöver utföra en uppgift med begränsade krav. Flera större program har dock utvecklats av större företag under långa tidsperioder, därmed har dessa program även ett större antal funktioner jämfört med exempelvis ett öppet programvaruprojekt utvecklat på fritiden. Dessa företag har även resurser att erbjuda stöd för kunder. Det finns exempel på öppen programvara som har ett fotfäste på marknaden och erbjuder tillräcklig funktionalitet i många fall, exempelvis Blender som används för att skapa 3D-grafik. Att uppnå denna position är dock ytterst svårt för mindre öppna programvaruprojekt, medan det inte är ekonomiskt lönsamt att öppna källkoden för större slutna projekt. Därmed kan man konstatera att öppen programvara i flera fall inte är ett alternativ för expertanvändare.

(TODO: bind ihop dessa stycken)

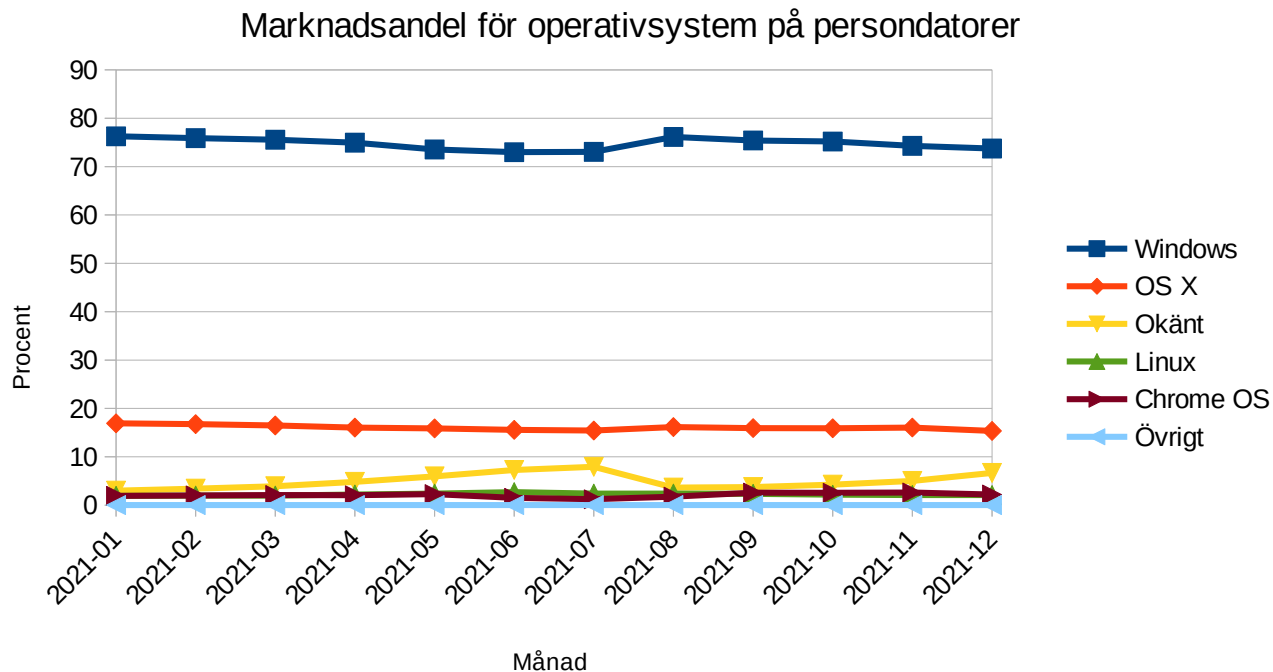
Några öppna programvaruprojekt uppmuntrar användare att betala för programvaran, men erbjuder sätt att utnyttja all funktionalitet kostnadsfritt. Ett exempel är operativsystemet elementary OS, som använder en "pay-what-you-want" modell. Genom att skriva in ett värde av noll hoppar man över betalningsprocessen och får en nedladdningslänk.[10]

Övriga projekt erbjuder grundfunktionalitet eller programvaran utan kostnad, men tar betalt för exempelvis molntjänster eller övrig funktionalitet. Ett exempel är lösenordshanteraren Bitwarden, som erbjuder standardfunktionalitet kostnadsfritt, bland annat ett obegränsat antal lagrade lösenord och anteckningar för dessa. Programvaran kan även köras på egna servrar i form av självhosting, men en del funktionalitet erbjuds endast mot betalning, såsom möjligheten att bifoga filer och dela privat data med exempelvis familjemedlemmar.[11][12]

Populär kostnadsfri programvara kan antingen vara öppen eller sluten. Därmed kan man konstatera att en del användare förmodligen använder populär öppen programvara, exempelvis OBS Studio på streamingtjänster, utan att vara medvetna om öppenheten.

3.3. Hårdvara

Konsumentfokuserad hårdvara innehåller ett förinstallerat operativsystem, som är optimerat att fungera väl på hårdvaran. Dessa operativsystem kan bestå av öppen programvara till en större grad, exempelvis Android, eller till största delen klassificeras som sluten programvara, exempelvis Windows. Dessa operativsystem innehåller i många fall slutna komponenter, såsom drivrutiner. (förklara, flera exempel)



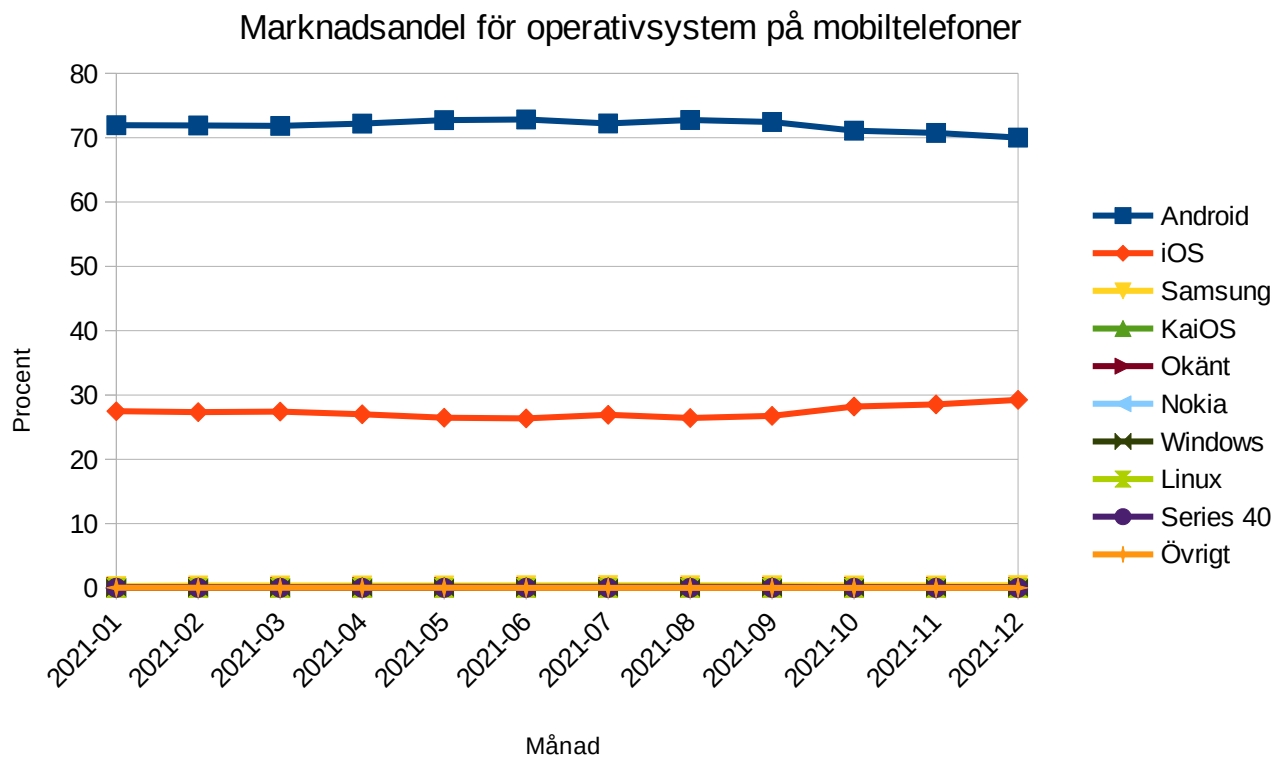
Figur 1: StatCounter (TODO: byt ut)

På persondatorer har Windows fortsättningsvis den största marknadsandelen av motsvarande operativsystem. Detta beror främst på att operativsystemet är förinstallerat på många datorer, vilket resulterar i att del populär programvara endast erbjuds för de största operativsystemen, exempelvis Microsoft Office. Hårdvaran optimeras för det förinstallerade operativsystemet, och kan kräva tilläggssarbete i form av baklängeskonstruktion (TODO: förklara) för stöd på mindre operativsystem. Många användare är bekanta med Windows, och har inte behov att installera övriga system. MacOS har en mindre marknadsandel jämfört med Windows, men hör också till förinstallerade operativsystem, i detta fall enbart på Apples konsumenthårdvara.

I allmänhet saknar vanliga användare större tekniska kunskaper. Installationsprocessen för ett operativsystem kan vara en främmande upplevelse. Det finns exempelvis ett flertal Linux-baserade operativsystem att välja mellan,

varav flera använder olika installerare med små skillnader, exempelvis Ubiquity för Ubuntu, Anaconda för Fedora, och en icke-grafisk installationsmetod för Arch Linux ...

(TODO: nämn Windows bootloader, problem med installerare)



Figur 2: StatCounter (TODO: byt ut)

På smarttelefoner är Android och iOS de största operativsystemen. En förutsättning för att mobiloperativsystem ska uppnå ett större antal användare är ett stort utbud applikationer, vilket minskar sannolikheten att nya operativsystem finner en långvarig plats på marknaden. Microsofts mobiloperativsystem, Windows Phone, misslyckades i längden trots stora

resurser och försök att locka större applikationsutvecklare till operativsystemet med hjälp av betalning.

Windows Phone och iOS samt hårdvaran operativsystemen kör på är i stort sett slutna. Android består av ett slutet OEM-lager på en öppen bas, tillgänglig under initiativet Android Open Source Project (AOSP). Källkoden tillgänglig i AOSP har använts för att skapa öppna varianter av Android, exempelvis LineageOS och GrapheneOS. Dessa alternativa operativsystem existerar dock inte för alla Android-enheter. LineageOS stöder officiellt 160+ enheter, medan GrapheneOS endast stöder 10. I några fall har dessa operativsystem inofficiellt stöd för Android-enheter, men kan medföra problem där delar av enhetens hårdvara inte fungerar väl med mjukvaran, exempelvis kameran.

I vissa fall låser exempelvis telefontillverkare starthanteraren på sina smarttelefoner, utan att erbjuda ett officiellt sätt att låsa upp den. En upplåst starthanterare krävs för att ersätta operativsystemet på enheten. Även då starthanteraren kan låsas upp, kan denna handling medföra problem där applikationer inte längre fungerar, exempelvis bankapplikationer. Detta beror på SafetyNet, säkerhetsfunktionalitet som Google implementerat för att kontrollera att enheten inte har modifierats. Dessa problem, kombinerade med varierande installationsinstruktioner för olika enheter, medför svårigheter då icke-expertanvändare vill använda ett öppet operativsystem på sina telefoner.

(TODO: nämn äldre hårdvara)

3.4. Användarupplevelse

Användarupplevelsen i öppen programvara kan variera mycket. I dagens läge sker kommunikation och kollaboration på populära onlinetjänster, såsom WhatsApp, Discord, Instagram, Zoom och Microsoft 365. Gemensamt för dessa

tjänster är att de är enkla att använda och erbjuder funktionalitet som majoriteten av användarna förväntar sig.

Att erbjuda samma upplevelse som dessa slutna tjänster är en utmaning för öppna programvaruprojekt. En stor del öppen programvara utvecklas på fritiden, och saknar därmed större arbetskraft och inkomster. Kommunikation och kollaboration över nätet kräver vanligtvis en eller flera centrala servrar som synkroniserar data mellan enheter. Behovet av flera servrar med större kapacitet ökar då antalet användare ökar, vilket i sin tur kräver större inkomster för att köpa hårdvaran och upprätthålla den. Dessa krav utesluter många mindre projekt. (TODO: ge exempel)

Onlinefunktionalitet krävs inte i all programvara, men stor konkurrens ställer krav på användarvänlighet för vanliga användare. God design av användargränssnitt kräver erfarenhet, vilket saknas i många öppna programvaruprojekt där källkoden betonas och arbetskraften är begränsad. Naturligt väljer användaren ett alternativ som är lätt att använda och förstå.

Som tidigare nämnts har expertanvändare specifika krav på funktionaliteten i programvaran, exempelvis en person som är grafisk designer eller VFX-artist. I vissa fall utgör denna programvara en industristandard, som är ytterst svår att ersätta med alternativ. Många av dessa program är slutna, delvis på grund av konkurrens.

(TODO: jämför program med bättre och sämre upplevelse, inkludera vanliga användare samt expertanvändare på något sätt)

3.5. Transparens och ideologi

En grundprincip i öppen programvara är transparens. Användaren har möjligheten att se hur programmet fungerar, och kan i flera fall hjälpa till att förbättra programmet utan större förhinder, exempelvis genom att rapportera fel eller föreslå ny funktionalitet. I allmänhet är många öppna programvaruprojekt relativt små, och erbjuder en större gemenskap jämfört med större, slutna projekt som utvecklas bakom stängda dörrar. Det finns även en vidsträckt förväntan att öppen programvara endast utför den uppgift användaren förväntar, utan skadlig funktionalitet som övervakning och annonser.

Att programvara är öppen medför inte nödvändigtvis en total brist på telemetri och annonser. Däremot anses öppen programvara vara ett alternativ för användare som oroar sig över personlig integritet, vilket kan leda till uppror om dessa förväntningar bryts. Ljudredigeraren Audacity är ett exempel på hur oförväntade förändringar och dålig kommunikation med användare kan leda till bakslag. Företaget Muse Group köpte programvaran och planerade att samla in information om användare, utan att diskutera dessa förändringar med användare.

För många vanliga användare är transparens inte en tillräckligt stor anledning att föredra öppen programvara. I slutändan används programvara för att utföra en viss uppgift, och populär programvara med god användarupplevelse och dokumentation betonas då. Slutna programvara är nödvändig för att exempelvis kommunicera med vänner och kolleger på WhatsApp eller Teams, eller använda Microsoft Excel på arbetstid.

4. Utvecklare

TODO: informationen är för grundläggande och kort, inkludera flera exempel

4.1. Introduktion

Termen utvecklare syftar på personer som skriver och upprätthåller källkoden i programvara. I vardagligt tal kan denna grupp även benämnas programmerare eller kodare. Detta avsnitt är uppdelat i olika huvudteman, som behandlar anledningar till att utvecklare skriver/inte skriver öppen programvara.

4.2. Kunskap, erfarenhet och personliga behov

Öppen programvara kan fungera som ett verktyg för att lära sig mer om ett visst område och utveckla sina kunskaper. I projekt med flera utvecklare kan man lära sig av andra, dels i att skriva kod, men även att använda olika verktyg och kommunicera med människor från olika delar av världen. Det är även möjligt att studera existerande kod i andra projekt, och utnyttja det inlärd i sina egna projekt.

Öppen programvara kan även användas för att förbättra sitt rykte som utvecklare. I allmänhet spelar goda och regelbundna kontributioner en stor roll i dessa projekt, då detta visar att man förstår hur projektet fungerar och är en pålitlig person. Om man saknar tidigare arbetserfarenhet, kan dessa kontributioner vara attraktiva på ett CV. Även personliga projekt kan användas för att bygga upp en portfölj.

Öppen programvara erbjuder även möjligheten att utveckla programvaran enligt personliga behov, exempelvis om funktionalitet saknas eller existerande funktionalitet kan förbättras. I några fall kan viljan att hjälpa andra betonas mer än personliga behov, särskilt om utvecklaren bidragit till projektet under en längre tid.

4.3. Gemenskap, nöje och ideologi

Öppen programvara betonar gemenskap. Syftet är att vem som helst ska ha möjligheten att använda och bidra till programvaran, vilket leder till att människor med olika bakgrund från olika länder arbetar tillsammans. Denna gemenskap kan leda till en vilja att hjälpa både användare och övriga utvecklare, samt att dela med sig kunskap.

För många utvecklare är utveckling av öppen programvara ett personligt nöje i form av en hobby.

4.4. Tidsbrist

...

4.5. Inkomst och företag

Företag kan anställa utvecklare i syfte att bidra till öppna programvaruprojekt. Vanligtvis sker detta till följd av att företaget utnyttjar denna programvara, och har särskilda önskemål på funktionalitet och prestanda.

5. Jämförelse och diskussion

...

Referenser

- 1: Fadi P. Deek, James A. M. McHugh, Open Source: Technology and Policy,
- 2: Arnoud Engelfriet, Choosing an Open Source License, 31 december 2009
- 3: , Desktop Operating System Market Share Worldwide, ,
<https://gs.statcounter.com/os-market-share/desktop/worldwide> (Hämtad 3 mars 2022)
- 4: Jonathan Corbet, Some 5.15 development statistics, 1 November 2021,
<https://lwn.net/Articles/874283/> (Hämtad 14 februari 2022)
- 5: , The GIMP Team, , <https://www.gimp.org/team.html> (Hämtad 28 februari 2022)
- 6: , How many people are there in Photoshop team that create this software?, ,
<https://feedback-readonly.photoshop.com/conversations/photoshop/how-many-people-are-there-in-photoshop-team-that-create-this-software/5f5f44f84b561a3d422ca524> (Hämtad 28 februari 2022)
- 7: Jonas Gamalielsson, Björn Lundell , Long-Term Sustainability of Open Source Software Communities beyond a Fork: A Case Study of LibreOffice, 2012
- 8: Tim Anderson, LibreOffice 7.2 brings improved but still imperfect Microsoft Office compatibility, 19 augusti 2021,
https://www.theregister.com/2021/08/19/libreoffice_72_native_for_apple/ (Hämtad 24 mars 2022)
- 9: , Open source alternatives to Adobe software, 29 december 2020,
https://www.reddit.com/r/Frugal/comments/kmewa4/open_source_alternatives_to_adobe_software/
- 10: Danielle Foré, Cheers to 10 Years, 31 mars 2021,
<https://blog.elementary.io/ten-years/> (Hämtad 14 mars 2022)
- 11: , Open Source | Bitwarden, , <https://bitwarden.com/open-source/> (Hämtad 13 mars 2022)
- 12: , Pricing for Business | Bitwarden, , <https://bitwarden.com/pricing/business/> (Hämtad 13 mars 2022)

TODO: lägg till alla referenser och snygga till