

TEORIBLOKK #1: Historie til og analyse av fenomenet "åpen kildekode programvare"

Kim Arild Moe

Tønnes Tobias Pedersen Røren

Vegar Sakseid

Introduksjon	1
Linux	2
Android og iOS	2
Copyleft	3
Utviklingsprosessen	4
Positive og negative sider med OSDM og CIPC	4
Konklusjon	6

Introduksjon

"Open Source Development Model" (OSDM) og "Conventional Industrial Product Creation" (CIPC) er to vidt forskjellige metoder som blir brukt i IT verden for å produsere et produkt.

CIPC er det vanligste og en metode som blir brukt ikke bare i IT verden men hos de fleste som produserer og selger et produkt. Her er det en eier som produserer et produkt som han selger til kunden. Denne modellen å produsere noe på har blitt brukt i alle tider og har lenge vært ansett som den eneste metoden som effektivt kan produsere noe på større skala.

OSDM derimot er oftes basert på frivillig arbeid der deltakerne ikke er del av en organisasjon, men deltar på et prosjekt frivillig uten å få kompensasjon for det.

Det er ikke før nyere tid med suksessen til Linux at dette har blitt ansett som en god måte å produsere et produkt.

Hva som er best av disse modellene er noe som har blitt diskutert mye etter suksessen til Linux og det er fremdeles stor uenighet rundt det. Vi skal se nærmere på disse og diskutere litt rundt ulemper og fordeler ved disse to metodene.

Linux

Linux er et av de mest kjente open-source prosjektene i verden. Prosjektet ble påbegynt av Linus Torvalds i 1990. Det var ikke meningen at det skulle bli noe stort

operativsystem, siden han bare utviklet det som en hobby. Men som vi alle vet, ble prosjektet han basen til et større open source basert operativsystem som har blitt videreutviklet av mange forskjellige personer. Det er estimert at kun 2 prosent av dagens linux base er skrevet av Linus Torvalds.

Det finnes mange forskjellige distribusjoner av Linux, hvor noen er open-source og noen er litt mer lukket. Linux lisensieringen sier at det ikke er lov til å bruke Linux basen til et lukket prosjekt. Men det finnes fortsatt noen litt mere kommersielle, lukkede distribusjoner hvor du får kunde hjelp og kan kjøre noen sertifiserte applikasjoner.

Linux er veldig kjent for måten de utvikler under open-source. Utviklingsmetoden deres blir ofte hyllet for å være mer effektiv og bedre enn flere closed-source utviklingsmetoder. Måten de utvikler er mye basert på open source prinsippene de står for. De bruker det store Linux samfunnet for kvalitetssikring og testing. Vanligvis ville man brukt slutt-brukeren som tester, men siden det er et så stort samfunn av utviklere som jobber sammen, kan de bruke hverandre.

Android og iOS

Hvis vi ser rundt i verden har vi et par store eksempler på OSDM og CIPC som alle kjenner igjen. Et av disse er i mobilmarkedet. De to største operativsystem for mobiler er nå Google sitt Android og Apple sitt iOS. Android som er en modell som er basert på Linux og følger en mer open source løsning hvor brukere kan lage apper og selge disse på Google Play til andre brukere. Apple igjen følger den standardmodellen som er mer lukket. Her kontrollerer Apple alt av apper som blir lagt ut i App Store hvor andre brukere kan kjøpe de.

Begge disse butikkene ble lansert i 2008 men Googles butikk har klarte å overgå Apple i forhold til antall tilgjengelige apper. I 2015 var det rapportert at Google Play hadde 70% flere app nedlastinger enn App Store i første kvartal av 2015, men Apples app inntekter var omtrent 70% høyere enn på Google Play.

En av grunnene til dette kommer av at Android er open source og gir mange “hobby” utviklere muligheten til å lage en liten app som de kan legge ut på Google play, mens Apple Store krever at appen skal godkjennes og testes av Apple. Dette krever selvsagt en mye høyere kvalitet av produktet før å få det godkjent for salg.

Dette igjen kan forklare litt av forskjellen på hvorfor inntekten på de to forskjellige butikken varier så mye. Utviklere som har lagt mye tid å penger ønsker gjerne å få noe tilbake, mens hobby utviklere gjør ofte dette ut av interesse å personlig læring og er ikke like interessert i tilbakebetaling. Dette fører ofte til at appene blir lagt ut gratis med eneste potensielle inntekt kommer fra reklame i appen.

En annen ting for potensielle utviklere å vurdere er at hvis du vil skrive en Apple-app, trenger du en Mac, mens Android-apper kan skrives på Windows, OS X og Linux.

Copyleft

Flere open source prosjekter bruker noe som heter copyleft. Dette er da isteden for copyright. Det garanter at brukeren har lov til bruke og endre programvaren slik de vil. Dette er for å beskytte open source prosjekter for copyright. Disse rettighetene blir beskyttet av flere forskjellige organisasjoner. Free Software Foundation er et eksempel på en organisasjon som beskytter og kultiverer open source programvare utvikling.

Utviklingsprosessen

Vi kan også se litt nærmere på selve utviklingsprosessen for disse to modellene.

I OSDM har vi ofte et samarbeid mellom mange utviklere som gjør dette på fritiden.

De kjenner ofte ikke de andre som jobber på dette prosjektet.

Her vil det ofte være at det er et system som er utviklet eller er under utvikling der en person ser en mangel eller mulighet for å oppdatere en funksjon. Denne personen vil da legge opp sin versjon av dette, hvor dette da vil bli en del av prosjektet.

Slike prosjekter kan ha få eller veldig mange personer som deltar. Som vi kunne lese i rapporten deltakelse i Apache (Inequalities in Open Source Software Development:

Analysis of Contributors Commits in Apache Software Foundation Projects) hvor 1-9-90 regelen så ut til å stemme med realiteten av deltagelse. Denne regelen sier da at kun 1% av medlemmene aktivt produserer innhold, 9% er generelt noe aktive, og de resterende 90% er passive "lurkers".

Positive og negative sider med OSDM og CIPC

Det positive med open source for programmering er at det er globalt. Du kan få personer med helt andre erfaringer og ideer til å se på et problem, de kan da komme med en unik og nytenkende løsning.

Det støtter også ideen om at mange hoder er bedre enn et. Her kan nesten et ubegrenset antall personer se på et problem og prøve å komme med en løsning siden det er åpent for alle.

Det er også gratis og i motsetning til CIPC produkter så eier du det, og kan derfor gjøre de forandringen du vil for at det skal passe deg.

Det er lett å starte opp en ide ettersom du ikke trenger finansiering. En person kan starte på det og andre kan bidra hvis de har interesse.

Men open source har noen negative sider også. Siden det er mange forskjellige som jobber på koden og kun skriver deler av funksjonaliteten, så er koden ofte ikke optimalisert for andre deler av programmet som har blitt programmert av andre.

Ettersom dette ofte bare er en hobby for mange så er et annet problem også at de ikke er langtids involvert for oppdatere kode og vedlikehold. Dette må da bli gjort av andre som kommer inn for å løse problemer når de skulle oppstå. Da kan problemer med hvor lesbar den originale koden er for andre som skal prøve å få den originale koden til å fungere med nye oppdateringer eller funksjoner oppstå.

At det er så mange som jobber på det og tilføyer nye funksjoner som de mener er nyttige fører ofte til at de blir komplekse å bruke og for personer uten en viss kompetanse kan det være umulig.

Ser vi på programmering og utvikling i CIPC så foregår dette ofte i selskaper og er veldig lukket. Det er ofte ulovlig å dele noe av koden med personer utenfor prosjektet. Her er det da kun en "liten" bestemt gruppe som jobber på prosjektet. Disse er ofte styrt av et hierarki basert ledelse hvor noen på topp bestemmer hvordan alt skal være og det er arbeiderne sin oppgave og oppfylle visjonen til lederne.

Her er også det viktigste konseptet at produktet skal kunne tjene inn mer en utgiftene for å produsere det.

Fordeler med dette er at firmaet ønsker å gjøre systemet sitt tilgjengelig for så mange så mulig, og det vil ofte være mye arbeid lagt ned i at det skal være lett å lære for personer uten kompetanse å bruke. Dette kan da gå på bekostning av funksjonalitet.

Firmaer har ofte en standard som utviklere jobber etter samtidig som de som er involvert ofte er mye i kontakt med hverandre. Dette betyr at om en skulle slutte så er det ikke så vanskelig for andre å ta over og jobbe videre på koden ettersom den følger en kjent standard.

Ettersom dette er et firma så har de ofte mye penger de kan bruke på å lage de beste løsninger eller ansette/leie inn personer som er de beste innenfor feltet.

Ulemper med CIPC er at firmaets høyeste mål er å tjene penger og ikke nødvendigvis lage det beste produktet.

Selv om de lager et produkt som kan brukes av de fleste så kan det være at de må kutte funksjoner for det ikke skal bli for innviklet. Dette kan føre til en begrensning av programmet.

Siden det kun er en bestemt gruppe som jobber på prosjektet kan de låse seg litt i gamle løsninger og ikke komme med nytenkning og heller bruke gamle sikre løsninger.

Produktet vedlikeholdes bare så lenge som firmaet anser det som lønnsomt

Vår Konklusjon

Hva som gir den beste programvarekvaliteten av OSDM og CIPC kan diskuteres, men vi ser klart at begge har noen ulemper og fordeler.

Det er også klart at dette ikke lenger er så svart/hvit som det var før. Mange av de store selskapene bruker open source i prosjektene sine. Det kan virke som disse to standard “metodene” blir mer og mer blandet.

Vi vil fremdeles si at CIPC leverer ofte de beste løsningene for folk flest. Dette er grunnet at CIPC sitt mål ofte er å selge til så mange som mulig. Derfor blir det ofte lagt mye tid på enkel installasjon (plug and play) og brukergrensesnitt slik at du skal kunne bruke basis funksjoner uten noen grunnkunnskaper. Men vi kan se negativ trend i samfunnet at det jevnlig skal komme en ny versjon av produktet som gjør at vi må oppdatere til den nye ettersom det gamle ikke vil bli vedlikeholdt lenger. dette er ofte drevet av penger og ikke nødvendigvis at det nye er mye bedre enn det gamle.

OSDM ser ut som den beste modellen for innovative ideer og nytenkning. Nye ideer og løsninger oppstår ofte her for så å bli tatt opp av et firma og tilpasset for folk flest. Dette tror vi er på grunn av at mange av disse prosjektene starter som interesser og drømmer hos privatpersoner/grupper og ikke er drevet av kravet om at de skal tjene penger på produktet. Men siden dette er produkter laget av personer som ofte har veldig høy kompetanse innenfor et felt kan det ofte bli vanskelig for “vanlige” forbrukere å bruke dette ettersom det ofte krever kunnskap for å installere/sette opp. det kan og skape problemer for andre og “hoppe” inn og begynne å delta pga kildekode fra OSDM-prosjekter være vanskeligere å lese for de som skal drive vedlikehold eller videreutvikling. Siden det er mange som jobber på prosjektet kan de ha ulik standard for hvordan de skriver kildekode og kommentarer, og det kan være vanskelig å sette seg inn i koden en skal jobbe med.

Vi kan se at disse to metodene å jobbe/produsere på blir mer og mer flettet inn i hverandre de siste årene. Skillene mellom dem blir mindre etter som open source prosjekter blir mer dratt inn i store firmaer.

