

## Eclipse: Godt skjult revolusjon

Det er klart vi stusset. Hele IT-verden stusset. Hadde IBM gått fra konseptene? Å først investere et uspesifisert antall timer og andre ressurser i utviklingen, for deretter å 'gi bort' hele stasen sammen med en startkapital på 40 millioner USD, var uhørt i 2001. Netscape hadde riktignok gjort noe lignende – med mer kode og mindre penger – 3 år tidligere, men mer i desperasjon enn av god vilje, og uten at resultatene ga grunn til applaus. IBMs valg ble dermed like oppsiktsvekkende som selskapet hadde håpet.

Siden har en lang rekke av de mest kjente kommersielle IT-aktørene kommet på lignende tanker, og vi hever ikke lenger et øyenbryn når et produkt plutselig blir 'open sourced'. Novell, Oracle, Sun, HP og flere har kastet seg på bølgen, og omfavner den alternative utviklingsmodellen med påtagelig energi.

### Trendsettende frigjøring

At IBMs åpning av utviklingsverktøyet Eclipse var trendsettende, hersker det med andre ord ingen uenighet om. Samtidig er det verdt å observere at i forhold til milliarder av USD som selskapet allerede hadde allokert til eller investert i Linux og andre Open Source-prosjekter, ble Eclipse nærmest for en pølse i slaktetiden å regne i økonomisk forstand.

*"Eclipse is a universal platform for integrating development tools."*

Om handlingen var trendsettende, var det få som levnet Eclipse særlige sjanser til å bli noe mer enn et komma i den store sammenhengen. Verktøyet var for ferskt til å ha noen stor kundekrets, og konseptet var annerledes enn tilsvarende for konkurrerende produkter. "Et publiserings-stunt fra IBM" var den gjengse kommentaren – og lite skjedde utover at utviklingen fortsatte. De samme arkitektene og utviklerne, ca. 40 i tallet og spredd over store deler av verden, arbeidet ufortrødent videre med store ambisjoner og ditto entusiasme. Resten av verden forble imidlertid skeptisk, ikke minst fordi Eclipse Foundation, stiftelsen som ble opprettet med fødselspengene, var kontrollert av IBM. Det fortonet seg som om IBM hadde gitt bort prosjektet til seg selv, og interessen fra andre leverandører ble deretter.

Dette forholdet forandret seg da Eclipse Foundation – med IBMs velsignelse – frigjorde seg fra opphavet i 2003, og åpnet for økonomiske bidrag, styreplasser og produkter fra hvem som helst. Eclipse ble til et 'fritt Open Source' prosjekt i stedet for bare et 'Open Source prosjekt'. Siden har utviklingen vært nærmest eksplosiv, hvilket er årsaken til at Eclipse i dag er et verktøy de fleste av oss bør ha grunnleggende kjennskap til.

### Hva er Eclipse?

Kortvarianten av svaret er – hentet fra stiftelsens Web-sider "a universal platform for integrating development tools". Karakteristikken gir en pekepinn, men yder teknologien ingen rettferdighet. Historien starter

med Java, som IBM var tidlig ute med å omfavne og som trengte utviklingsverktøy. Mens Sun vinglet, og ikke helt forsto hvilket gullegg de hadde i hendene, satte IBM like godt i gang på egen hånd – med blant annet det som senere ble til Eclipse.

### **Født med Java i munnen**

Java er med andre ord en viktig del av verktøyets historie og praktiske rolle, og en av årsakene til at Eclipse ikke bare eksisterer, men vokser i popularitet over alle forventninger. Flere av verdens ledende Java utviklingsmiljøer karakteriserer Eclipse som det største (i utbredelse og anvendelse) og viktigste Java-verktøyet på markedet.

Faktum er at også det motsatte kan hevdes. Eclipse er en viktig årsak til Javas popularitet i utviklingsmiljøer. Slike miljøer er alltid sensitive overfor tilgjengeligheten av effektive verktøy, til åpenhet og til mulighetene for å dele kode og komme raskt til målet. Eclipse har hatt viktige bidrag å komme med i den forbindelse – også før det egentlige frislipet. Verktøyet har – med god hjelp fra IBM – kompensert for tallrike pussigheter fra Java-eier Sun, som gang på gang har kommet på kant med nettopp utviklingsmiljøene. Enkelte hevder sågar at uten IBM og Eclipse ville Java ha vært en ubetydelighet i dag, ikke minst på grunn av Suns lite heldige omgang med åpenhet, lisensiering, standardisering og verktøykvalitet.

Hvorvidt en slik påstand er riktig eller rimelig, spiller mindre rolle i dag. Det faktiske forhold er at programmeringsmiljøet Java er langt mer utbredt enn for eksempel Windows, og representerer en av de viktigste utviklingsplattformene i dagens marked.

Historien indikerer en tett kobling mellom Java som programmeringsmiljø og Eclipse som utviklingsverktøy. Denne koblingen er et faktum i og med at Eclipse for det første er skrevet i Java, og for det andre leveres med Java-verktøy som er blant markedets beste. Eclipse er imidlertid langt mer enn et Java-basert utviklingsverktøy for Java-utviklere. Eclipse er universelt rammeverk – en plattform – for programmeringsverktøy, med en 'plugin'-basert arkitektur som inviterer til å plugge inn verktøy av alle tenkelige slag. Den fleksibiliteten en slik åpenhet gir, er årsaken til at de fleste store aktører i programvare-segmentet i løpet av det siste året har signalisert eller etablert et aktivt forhold til nettopp Eclipse.

At Sun og Microsoft – enn så lenge – er fraværende i denne klubben, er knapt overraskende. Begge har verktøy som konkurrerer direkte med Eclipse, og tviholder på oppfatningen av at selvgjort er velgjort. Microsofts Visual Studio har sågar tallrike arkitektoniske fellestrekk med Eclipse, men hemmes av lukkede grensesnitt som gjør det vanskelig for andre å plugge seg inn i rammeverket. For Suns vedkommende er NetBeans det magiske ordet, sågar en Open Source satsing som teknisk sett kan bli en verdig konkurrent til Eclipse, men som – gitt Suns historie – har tungt for å få massefart i utviklingsmiljøene. Som tilfellet var for Eclipse frem til 2003, er NetBeans hemmet av å ikke være 'fri Open Source', kun 'Open Source'.

Uansett er denne konkurransen positiv, og sjansen er overveiende for at alle tre lever videre i overskuelig fremtid med et eller annet styrkeforhold, der Eclipse beholder sin overhånd – av årsaker som har fint lite med Java og historie å gjøre.<sup>5</sup> Det er nettopp disse årsakene som gjør Eclipse verdig en gjennomgang her i Mellvik-Rapporten.

### En bemerkelsesverdig arkitektur

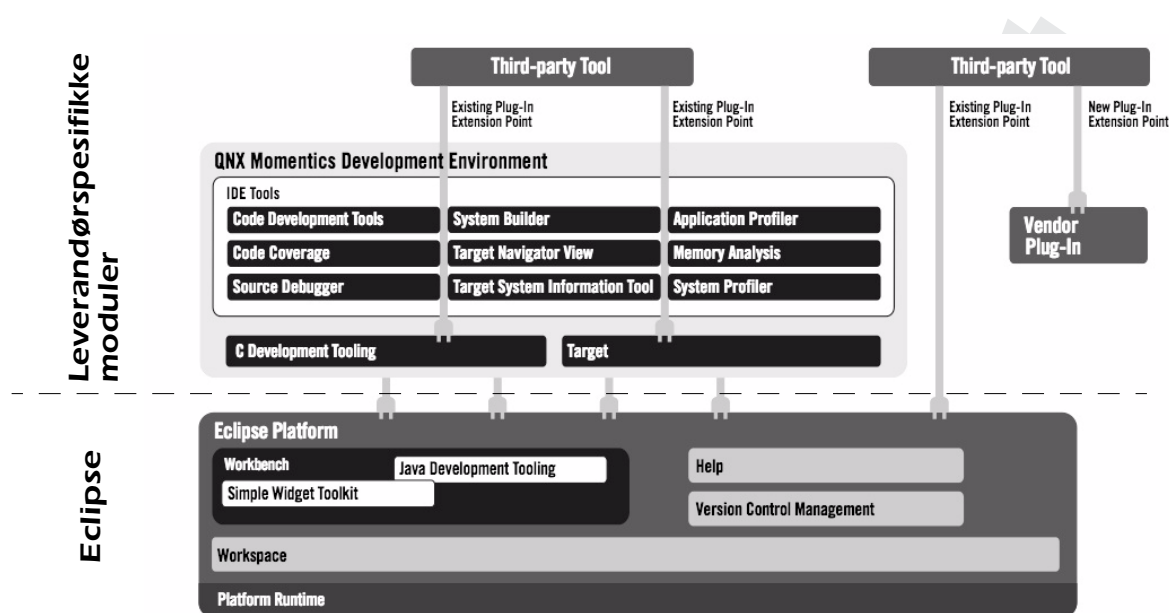
Ved første øyekast fremstår Eclipse som en hvilken som helst moderne plattform for programvareutvikling – med komponenter som intelligent teksteditor (som gjenkjenner programmeringsspråk og hjelper til med syntaks, biblioteker etc.), debugger, hjelpesystem, revisjonskontroll, samarbeidsmekanismer, arbeidsbenk, testmiljø og så videre. Mens disse elementene er viktige – og av høy klasse i Eclipse, er det *plugin*-arkitekturen som gjør plattformen unik.

Arkitekturens åpenhet og ryddighet, kombinert med plattformens leverandøruavhengige Open Source-baserte fundament, sørger for det første for at konkurrerende leverandører kan samarbeide uten kontinuerlig å måtte kontrollere at ingen andre tar seg utilbørlig til rette. Effekten er nøyaktig den samme som vi har sett for Linux, men enda mer tydelig i og med at interessen først tok av etter frikoblingen fra IBM. For det andre stimulerer arkitekturen til utvikling av små og store moduler som i sin tur plugges inn og sømløst finner seg til rette i omgivelsene. Et formidabelt tilfang av slike moduler med funksjoner som strekker seg langt forbi tradisjonell utvikling, har dukket opp i løpet av to hektiske år, og strømmen fortsetter å tilta.

En tredje faktor er at *plugin*-arkitekturen gjør det enkelt for leverandører å velge lisensieringsmodell. Siden en *plugin*-modul er en separat enhet og ikke automatisk blir en del av den Open Source-baserte plattformen, kan modulene betraktes som selvstendige programmer i juridisk forstand, med den følge at grensene mellom Open Source og kommersiell (*closed source*) programvare er ryddig og konfliktfri. En rekke kommersielle utviklingsverktøy med Eclipse som fundament har kommet på markedet de siste 18 månedene, med Momentix fra QNX (figur 1) som godt eksempel. Leverandører av slike produkter er uten unntak aktive bidragsyttere i videreutviklingen av Eclipse.

Og sist, men ikke minst er det naturligvis av stor betydning for populariteten at Eclipse kan lastes ned uten kostnader ([www.eclipse.org](http://www.eclipse.org)), og umiddelbart representerer et glimrende utgangspunkt for både eksperimentering og utvikling av applikasjoner i Java, C, C++ og andre språk – gjerne flere av dem i samme applikasjon. Noen ulempe er det heller ikke at Eclipse er plattformuavhengig og kan kjøres på Windows, Linux, Solaris, Mac OS X og flere andre plattformer. Riktignok går ikke alle moduler på alle plattformer, men med tilgjengelig kildekode er tilpasning til plattformer ofte kun et spørsmål om tilstrekkelig interesse og tid.

<sup>5</sup> Ferske toner fra Microsoft – som vi skal komme tilbake til i neste utgave, indikerer at sannsynligheten for at selskapet involverer seg med Eclipse i løpet av det neste året, er vesentlig større en tilsvarende fra Sun.



**Figur 1** Figuren viser hvordan Eclipse brukes som plattform for leverandørspesifikke utviklingsverktøy, i dette tilfellet QNX Momentix. QNX vektlegger C, C++ og Embedded C++ som programmeringsspråk i tillegg til Java, og poengterer plattformuavhengigheten – for utviklingsverktøyet og den resulterende koden.

### Fristende allsidighet

Fortsatt handler det imidlertid om tradisjonell programvareutvikling – riktignok med en uvanlig og åpen arkitektur, men hva er det som gjør Eclipse interessant også utenfor tradisjonelle utviklingsmiljøer?

Åpenhet skaper kreativitet og elegante arkitekturer stimulerer den samme kreativiteten. Eclipses fleksibilitet og tilgjengelighet har tydelig stimulert tallrike kreative sjeler til å tenke annerledes, med den følge at moduler for utvikling av helt andre ting enn programvare har dukket opp. Først som eksperimenter og demonstrasjoner av hva som er mulig, senere i mer seriøse sammenhenger.

IBM, som fortsatt har et av de største brukermiljøene for Eclipse, har kanskje det beste eksemplet i det de kaller *Eclipse Rich Client Platform*: Et rammeverk for bygging av plattformuavhengige brukergrensesnitt som blant annet benyttes i selskapets *Workplace Client Software*, og som potensielt kan erstatte Windows eller andre brukergrensesnitt som arbeidsmiljø. Med den betydningen begrepet operativsystem har fått i våre dager, med hovedvekt på brukergrensesnitt i stedet for tradisjonelle OS-funksjoner, kan Eclipse i voksende grad fortone seg som et operativsystem – ifølge enkelte eksperter.

CA (Computer Associates), en relativ nykommer i Eclipse-sammenheng, har ambisjoner om å utvikle et rammeverk for systemadministrasjon under Eclipse. En annen fersking i dette selskapet er BEA, som bidrar med kode og utviklere til et prosjekt som utvikler en plattform for Web-verktøy. Selskapet har båret Java-fanen høyt siden midt på 90-tallet og har holdt avstand til Eclipse for å tekkes Sun. Med den massefart Eclipse har fått, er det imidlertid ikke lenger noen opplagt

fordel å holde seg til Sun for å delta i videreutviklingen og standardiseringen av Java. "Det går for sakte" sier BEA, med referanse til Suns Java Community Process.

En annen gruppe bruker Eclipse som rammeverk for verktøy som 'graver' i datavarehus og genererer rapporter (BIRT, *Business Intelligence and Reporting Tools*), mens en tredje konsentrerer seg om anvendelser for helsemarkedet, der behandling og håndtering av data skal tilfredsstille helt spesielle krav. Og sist, men ikke minst finnes det en rekke eksempler på at Eclipse benyttes som skreddersydd prosjektstyringsverktøy.

Det er allsidigheten som gir Eclipse revolusjonære tilbøyeligheter. Selve plattformen, gjerne uten de Java-relaterte utviklingsmodulene, utgjør i seg selv et attraktivt rammeverk for utallige anvendelser. Utviklere finner at de kommer både raskere til målet og får et bedre resultat gjennom Eclipse enn ved å finne opp kruttet på nytt. Tankegangen er på mange måter den samme som for vindussystemer i sin alminnelighet, men vesentlig enklere både for utviklere og brukere, ikke minst når flere applikasjoner skal samarbeide.

## Oppsummering

Eclipse er ikke bare blitt markedets mest populære utviklingsplattform for Java på rekordtid, men gjør også vei i vellinga for andre programmeringsspråk. Dessuten – og her er årsaken til at Eclipse er vel verdt å legge merke til – eigner plattformen seg til formål som strekker seg langt utenfor tradisjonell programmering og programvareutvikling.

Massefarten fortsetter å øke. I tillegg til CA og BEA, har Borland, Sybase og i noen grad Oracle nylig hoppet på Eclipse-toget. Vi observerer sammenhenger som gjenkjennes fra Linux: Plattformen er fri for kommersielle interesser og dermed ufarlig sett fra leverandørenes side, som alle har den samme mulighet til å redusere kostnader og samtidig få bedre produkter.

Konsekvensene er uoversiktlige. Eclipses natur som plattform der nærmest hva som helst kan bygges, sørger for at det er utviklernes kreativitet som setter grensene, ikke verktøyet. Dette er en drømmesituasjon for enhver utvikler, og det sikreste vi kan si om fremtiden er at hva vi har sett så langt kun er begynnelsen. Som Borlands Raaj Shinde sier det: *"The architecture is incredibly elegant!"*

Så spørs det hvor lenge Sun og Microsoft tør å stå utenfor. For hver eneste måned som går, havner de lenger på etterskudd – i en nisje som kan vise seg å bli et segment eller et helt marked, kanskje sågar en farlig konkurrent? ■