

Hvor enkelt kan det bli?

“Så enkelt at selv mor kan gjøre det” het det før likestillingen brakte fars inkompetanse frem i lyset. I dag er det lov for alle å innrømme sine svakheter. Ingen er best i alt. Almennelige verktøy, spesielt IT-verktøy, tar hensyn til dette.

Pek-og-klikk er et ektefødt barn av tankegangen: Det skal bli så enkelt at alle kan gjøre det. Resultatene er bemerkelsesverdige – på mange måter. På den positive siden er det åpenbart at terskelen for bruk av til dels sofistikerte verktøy er senket dramatisk, hvilket har gitt store gevinster i mange sammenhenger.

Men medaljen har en bakside: Det finnes grenser for hvor langt forenklingen kan gå og fortsatt være meningsfylt. Faktum er at overforenkling kan være vel så negativt som forenkling kan være positivt. Det skal både skjønn og innsikt til for å vite hvor grensene bør gå. Slik innsikt er mangelvare i de fleste bransjer og sammenhenger – inklusive IT-bransjen. Konsekvensene er like interessante som de er fryktingytende.

Det mest nærliggende eksemplet på dette tveeggede sverdet er tekstbehandling. Visst er det blitt trivielt for praktisk talt hvem som helst, med minimal opplæring, å benytte Word, WP, OpenOffice eller lignende verktøy til skriving av enkle dokumenter. Programmene er avanserte skrivemaskiner som er lette å bruke når tastaturskrekken er overvunnet og noen grunnleggende konsepter forstått. Avstanden derfra til å kunne sette opp dokumenter og maler, formatere tekst og gjøre bruk av sofistikert typografi, er imidlertid lang. I beste fall kan én av tusen brukere gjøre en fornuftig jobb på et slikt nivå, hvilket er naturlig, ikke kritikkverdig. Det som er kritikkverdig er å ha forventninger om at avanserte håndverksfag skal kunne utøves med god kvalitet fordi verktøyet er enkelt å bruke. Det skal forståelse, interesse, erfaring og kunnskap til for å utøve et hvilket som helst håndverk, inklusive typografi.

Mens tekstbehandling er et relativt ufarlig eksempel fordi resultatenes kvalitet er lett synlige, er forholdet et annet for programmering. Avanserte, ‘visuelle’ programmeringsverktøy bygget rundt forhåndsdefinerte maler og pek-og-klikk grensesnitt, kan se ut som enestående effektivitets-BOOSTERE, men er i mange tilfeller det motsatte. Slike verktøy skaper illusjoner som det er vanskelig å argumentere mot, fordi “WHAT YOU SEE IS NOT WHAT YOU GET”. Resultatet er gjerne funksjonelt innenfor visse rammer, og kan påviselig gjøre en jobb. At det som befinner seg under overflaten, gjerne er av elendig kvalitet og mangler robusthet – for eksempel feilhåndtering, datakontroll og konfigurasjonsmuligheter, kommer først for en dag etter at løsningen er tatt i bruk.

Avanserte og effektive verktøy er en viktig del av IT-hverdagen. Å forenkle og effektivisere utviklingsprosessen er en nødvendig oppgave som mange verktøyleverandører tar på alvor. Det er ikke verktøyene i seg selv vi er ute etter i denne sammenheng, men måten de selges på og illusjonene som skapes. Et håndverk kan effektiviseres gjennom avanserte verktøy og tek-



Mellvik-Rapporten® utkommer 11 ganger i året og utgis av:
Team Mellvik as
Postboks 54 Holmenkollen
NO-0712 Oslo
Telefon 22 14 26 47
Telefaks 22 49 35 98
Org.nr. NO 966989351 MVA

Ansvarlig redaktør:
Hanne Mellingen
Fagansvarlig:
Helge Skrivervik
Korrektur:
Kari Mellingen

Epost: info@mellvik.no
URL: www.mellvik.no
ISSN 0804-9386

Særtrykk tilbys, ettertrykk og kopiering forbudt.

Se baksiden for informasjon om abonnement og bestilling av tidligere utgaver.

Mellvik-Rapporten er et registrert varemerke tilhørende Team Mellvik as.

niske hjelpemidler, men blir ikke dermed mindre krevende. Å gjøre en god jobb krever fortsatt den samme innsikt og evne til å forstå problemene som skal løses. Verktøyene gjør jobben enklere, kvaliteten sikrere og veien til målet kortere – for en dyktig håndverker. Avanserte verktøy i hendene på inkompetente brukere er like farlig uansett sammenheng og bransje. Like lite som en spikerpistol eller en avansert sag hører hjemme i hendene på et barn, har et avansert programmeringsverktøy noe å gjøre i hendene på amatører.

Programmering og systemutvikling krever fagkunnskap, erfaring – og avanserte verktøy. Historiene om kostnadssprekk og andre katastrofer i kjølvannet av det vi noe stereotypet kan kalle 'ivrige 19-åringer med VB under armen', er for mange til å ignorere. Det er dyrt å være fattig, og fagkunnskap koster. At kunnskapsløshet koster enda mer kommer for dagen først i etterhånd.

Å 'operere' en bil er rimelig ukomplisert, mens å kjøre den i et dynamisk trafikkbilde er krevende for hvem som helst. Derfor må vi igjennom en prøve før vi slippes løs på veiene – blant annet fordi konsekvensene av å ikke beherske verktøyet lett blir dramatiske. Hva kan det komme av at vi er mindre opptatt av konsekvensene av dårlig programmering? Visst er de livstruende. En bil som får innmontert en feil del, en tannlegepasient som får feil bedøvelse, en kran som ramler over fordi vekt opplysningene er feil – og så videre: Forskjellen er at det er mindre opplagt hvor feilen sitter, og dermed hvor ansvaret burde være. Vi skylder på datafeil, mens den egentlige årsaken gjerne er manglende gangsyn i forbindelse med tildeling av programmeringsjobber. At en person har universitetsutdannelse, Datakortet eller en MSC-diplom i baklommen er ikke det samme som at vedkommende er en god programmerer eller systemutvikler.

Det svakeste punktet er ikke overforenklende verktøy eller selvlærte programmerere, men tildeling av jobber uten reell evaluering av kompetanse. Det er et paradoks at vi er mer kritiske overfor elektrikere, rørleggere og snekkere enn overfor programmerere.

Oslo, 5. mai 2003

