

Egendeklaration av Programvaror

Svensk metod för kvalitetsmärkning av
Programvaror med internationella
standarder som grund

Bertil Håkansson



Via
teldok.

39

TELDOK visar hur IT används och dokumenterar erfarenheter av tidig användning. Med stöd från Telia bidrar TELDOK till: dokumentation, publicering och spridning, studieresor och konferenser.

Hittills har TELDOK finansierat, publicerat och distribuerat ca 200 rapporter som beskriver, och/eller ger bakgrunden till, tidig användning av ny informationsteknik.

En glimt av TELDOKs senaste utgivning finns längst bak i denna rapport. Rapporterna sprids gratis till ca 5.000 personer som bett att få dem kontinuerligt.

Utgivna TELDOK-rapporter kan beställas på 020 23 00 11 eller www.teldok.org. Ange rapportnummer! Rapporterna är gratis i enstaka exemplar, högst fem titlar per beställning; beställning av fler faktureras.

Ytterligare information lämnas gärna av ledamöterna i den brett sammansatta TELDOK Redaktionskommitté:

- Bertil Thorngren (ordf), CIC vid Handelshögskolan, dbt@hhs.se
- Anna Karlstedt, IMIT, 08 736 9471, FAX 08 32 65 24
- Göran Axelsson, Statskontoret, 08 454 4690
- Hans Iwan Bratt, ITfacts, 08 753 1851, 070 779 7796
- Birgitta Frejhagen, InfoKomp AB, 08 725 8700
- Kurt Gladh, Dataföreningen, 08 506 40 400
- Johan Jörgensen, Result Ventures, 070 491 0381
- Eva Lindencrona, EU, eva.lindencrona@cec.eu.int
- Lennart Ohlsson, IM Venture & Finans AB, rintec@ibm.net
- Agneta Qwerin, RSV DataService, 08 764 8378
- Hasse Samuelsson, Svensk Industriförening, 08 440 1170
- Bengt-Arne Vedin, Metamatic AB, 08 661 2818
- Lotta Åkerblom, Growinlife Sweden AB, 08 458 1606
- PG Holmlöv (sekr), Telia Research och CIC, pg@stones.com, 08 713 5129

teldok.

© 2000 författaren Bertil Håkansson (bertil.hakansson@infocomab.se), 08 86 54 00, och TELDOK var för sig

TELDOK uppmuntrar till eftertryck för enskilt bruk, med angivande av källa. Kommersiell vidare spridning ej tillåten utan överenskommelse med TELDOK eller författaren

Formgivning: Tommy Gripstad, Ateljé Faktor, Stockholm 2000

Omslag: PG Holmlöv, pg@stones.com

Digitalisering: Mart Marend, www.books-on-demand.com

Offsettryck: Nykopia Tryck AB, Stockholm 2000

<http://www.teldok.org/>

Innehåll

Företal	5
1. Metod för bättre programvarukvalitet	7
Metod för bättre kvalitet	8
Att tillfredsställa behov och förväntningar	8
Kvalitet från början	9
Gemensamt kvalitetsbegrepp	10
Öppenhet ger trovärdighet	11
Kvalitetsmärkning ger många fördelar	12
Egendeklaration – ett första steg	13
Märke som ger exportintäkter	13
TCO '92 – Sveriges mest spridda varumärke	13
2. ISO-standarder som grund för utvärdering av programvaror ..	15
Mjukvarukvalitet	15
De sex huvudegenskaperna	17
Underegenskaper och attribut	18
Användarkvalitet	19
Tre steg till kvalitet	20
Livscykelkvalitet	21
Att genomföra utvärderingen	22
Utvärderingskrav	24
Utvärdering i fem steg	25
Utvärderingsrapport	25
Kostnader	26
Objektivitet och jämförbarhet	26
3. Egendeklaration bygger på regler och öppenhet	28
Objektivt och trovärdigt	28
Öppenhet avslöjar fusk	29
Konkurrenterna skärper kraven	30
Testat system	31
Kontrakt med SPI	32

Projektgrupp	33
Fem utvärderingsaktiviteter	34
Utvärderingsnivå	34
Utvärderingsplan	36
Functionality	37
De viktiga attributen	37
Kravnivåer	40
Metoder	43
Metod O. Inspektion av dokumentation	44
Metod G. Inspektion av systemlösning	45
Två dagars testning	46
Utvärdering i grupp	47
Fortlöpande redovisning	48
Utvärderingsverifikat	48
Metod T. Grundfunktioner	49
Indata till utvärderingen	50
Rådata	51
Resultat av utvärderingen	52
Utvärderingsresultat för ZoomON Z2	53
Minst 10+12 godkända attribut	54
90% attribut godkända	55
Principer för SPI 2000 och egendeklarationer	56
4. Tre företags erfarenheter av utvärderingen	58
5. Möjligheter inför framtiden	63
Kvalitetssäkring i andra länder?	63
Konferenser om certifiering	64
EUREKA	64
ESPRIT	65
Dansk metod enligt ISO/IEC	65
Privatcertifiering i USA	66
Framgångsfaktorer	67
Bundsförvanter	68
Vidareutveckling	68

Företal

Människans förhållande till datorer har nog alltid kännetecknats av frustration över att datorerna inte beter sig som väntat. Men datorprogram är skapade av människor. Ingen människa uppfattar en situation eller händelse på exakt samma sätt som en annan människa. Det som är självklart för den ene är obegripligt för den andre.

Detta gäller också programmerare, som anser det självklart att användaren skall göra som han/hon hade tänkt sig.

Men nu tänker de flesta annorlunda.

Därför måste programmeraren förmås att se sitt program från användarperspektiv och anpassa din produkt till hur de flesta faktiskt agerar i kontakt med programvaran.

Bristen på sådan anpassning kostar näringslivet miljarder årligen i väntetid och konsultarvoden.

Detta är bakgrunden till initiativet från Föreningen Svensk Programvaruindustri, SPI att skapa en varudeklaration för programvara Spi2000.

Den här skriften vill informera om hur en arbetsgrupp inom SPI med hjälp av internationella standarder för utvärdering av programvara har tagit fram ett system för kvalitetsmärkning, som troligen är den första i sitt slag i världen. Den beskriver hur de första utvärderingarna enligt Spi2000 av programvaror på den svenska marknaden gick till i praktiken och erfarenheterna av detta.

Bertil Håkansson, som är författare till rapporten, har med sin gedigna bakgrund i branschen lyckats förklara ett komplicerat ämne på ett lättillgängligt sätt. Det är TELDOKs förhoppning att programtillverkare genom att läsa rapporten ser fördelarna med en testad programvarukvalitet och att användare lär sig vilka krav man kan ställa på sin leverantör att dokumentera programvarans egenskaper genom standardiserade tester. Nu när det finns metoder för att varudeklarerat programvaror finns ingen anledning längre att köpa grisen i säcken.

Hasse Samuelsson

hasse.samuelsson@sinf.se, ledamot TELDOK Redaktionskommitté

1. Metod för bättre programvarukvalitet

Är det möjligt för en programvarutillverkare att själv testa och där-
efter avge något objektiva omdöme på sin programvaras kvalitet
och funktion? Skulle detta vara av intresse för köpare och använ-
dare att ta del av en sådan deklARATION? Är den tillförlitlig? Skulle program-
varans kvalitet med en sådan metod kunna förbättras? Behövs det över
huvud taget bättre kvalitet på programvarorna?

På den sista frågan kan man nog tveklöst svara ”ja”! Bara det faktum att
så många människor i sitt dagliga arbete tvingas starta om sin dator då och
då därför att den hängt sig eller något annat fel uppstår, antyder att det är
något fel i programvarorna eller att fel uppstår mellan programvarorna.
Sådant orsakar årligen stora tidsförluster.

Runt om i världen finns det felaktiga eller ofullständiga programvaror
som orsakat och orsakar stora, extra kostnader. Ett exempel från den ame-
rikanska delstaten Colorados huvudstad Denver kan belysa detta.

Där hade man i mitten på 1990-talet byggt en ny, stor internationell
flygplats. Den är tio gånger bredare än Heathrow utanför London och tre
stora jetflygplan kan landa samtidigt, t o m i dåligt väder.

Flygplatsen försågs med ett automatiskt transportsystem för hantering
av allt bagage. Kostnaden uppgick till 193 miljoner dollar. I priset ingick
4 000 datorstyrda vagnar som skulle betjäna alla ankommande och avgående
flygplan. Vagnarna manövrerades med hjälp av cirka 100 datorer, 5 000
elektriska ”ögon”, 400 radiomottagare och 56 streckkodsläsare. Alltsam-
mans styrdes och övervakades av en programvara som skulle se till att allt
bagage togs om hand och distribuerades till och från alla berörda flygplan.

Tyvärr fungerade inte systemet som det var tänkt. Hela invigningen av
flygplatsen fick skjuts upp, först en månad, sedan ett halvt år och sen ytter-
ligare några månader. Fördröjningen medförde extrakostnader på över en
miljon kronor – per dag. Orsaken var främst en programvara som inte höll
mättet.

Programfel och undermåliga programprodukter kostar mycket stora belopp i stillestånds- och konsultkostnader. Hur mycket vet vi inte, men det är inte ovanligt med kostnader på mellan 200 000 och 300 000 kronor per arbetsplats för att åtgärda programfel eller ofullständigheter. Det är inte acceptabelt.

Att höja programvarornas kvalitet och förbättra deras funktioner skulle därför kunna spara mycket pengar, tid och frustration. Frågan är förstås hur detta ska göras.

Metod för bättre kvalitet

Denna skrift berättar om en metod som gör det möjligt att genom utvärdering och certifiering av programvaror successivt höja kvaliteten och förbättra funktionen på programvaror. Certifiering av programvaror innebär att man genom ett testförfarande visar att den testade produkten uppfyller definierade funktions- och kvalitetskrav.

Att en programvara ska fungera borde vara självklart och själva begreppet kräver inga större utläggningar. Lite knepigare är det kanske med ordet ”kvalitet”. Vad avser vi med kvalitet när det gäller programvaror? Och vem ska bestämma vad kvalitet är?

Frågan om vad som menas med kvalitet är nästan lika gammal som människan och har diskuterats såväl under antiken av bl a Demokritos och senare av andra storheter som exempelvis Galilei, Descartes, Newton och Locke. I dag har kvalitet blivit allt viktigare i en mängd sammanhang.

Själva ordet ”kvalitet” kommer från det latinska ordet ”qualitas” och står för ”beskaffenhet”, ”egenskap”. ”Qualitas” härstammar i sin tur från ”qualis”, som kan översättas med ”på något sätt beskaffad”.

Intressant i sammanhanget är också Nationalencyklopedins resonemang under uppslagsordet ”kvalitetsteknik”, där man bl a kan läsa följande:

Att tillfredsställa behov och förväntningar

”Synen på begreppet kvalitet har under senare år ändrats. Tidigare betydde kvalitet enbart ’uppfyllande av specifikationer’, men i dag definieras en varas eller tjänsts kvalitet som dess förmåga att tillfredsställa kundernas behov och förväntningar.” – ”En lyckad kvalitetsstrategi bygger på att man sätter kunderna i centrum, baserar beslut på fakta och engagerar samt-

liga anställda i ett ständigt arbete med att förbättra företagets alla processer, inom såväl utveckling och administration som tillverkning.”

Lite längre fram kan man under samma ord läsa: ”Utvecklingen under de senaste årtiondena har lett till att en allt större del av kvalitetsinsatserna genomförs allt tidigare under framtagning av produkter och processer.”

Av dessa citat framgår att kvalitet i högsta grad angår kunden, slutanvändaren. Denne köper exempelvis en programvara för att med dess hjälp kunna utföra vissa saker på det sätt han förväntar sig eller på det sätt som programvaran, enligt försäljaren, kan utföra olika uppgifter. Kundens förväntningar på vad programvaran kan göra är således ett kvalitetsbegrepp.

I förlängningen av detta resonemang kan man emellertid också lätt inse att kvalitet även är viktigt för den som marknadsför programvaran. Säljer han eller hon en produkt som enligt köparen inte håller den kvalitet som utlovats eller som köparen förväntar sig, då hamnar säljaren snart i onåd och får svårare att sälja sina produkter. Även om företaget senare kommer fram med bättre produkter är det risk för att kundens tidigare erfarenheter av undermåliga programvaror ändå besudlar företagsnamnet under längre tid.

Kvalitet från början

Av citaten från Nationalencyklopedin lär vi oss även att kvalitet uppenbarligen måste byggas in i produkten från allra första början. Utvecklaren, eller den som ansvarar för framtagningen, bör därför under utvecklingens gång kunna kontrollera kvaliteten så att slutprodukten ur kvalitetssynpunkt blir vad han tänkt sig. Att försöka rätta till kvaliteten i efterhand är betydligt besvärligare och mer kostsamt än om man kan bygga in rätt kvalitet från allra första början. Detta innebär i sin tur att även den som utvecklar produkten har ett behov av att på något sätt kunna mäta kvaliteten.

Samma behov att mäta och redovisa kvalitet bör finnas inom hela programvaruindustrin.

Kunde man få till stånd en vederhäftig utvärdering, märkning och redovisning av funktion och kvalitet på programvaror, skulle det kunna ge klara fördelar. Köparen skulle känna sig säkrare på den programvara han köper och säljaren skulle bättre kunna argumentera för den kvalitet som programvaran bevisligen har.

Särskilt mindre och medelstora programvaruföretag kan ha svårt att hävda sig på en internationell arena, där de stora elefanterna dansar så tungt och trumpetar så högljutt att de lätt överröstar de små företagens

svagare marknadsröster. Om man däremot kan etablera och arbeta in ett trovärdigt utvärderingsförfarande skulle sannolikt flera köpare våga köpa programvaror även från mindre företag.

Antag att en köpare har för avsikt att köpa en viss programvara med vissa funktioner. Varje programvaruleverantör beskriver på sitt sätt respektive programvaras förtjänster. Det gör inte valet alltför lätt. Om några programvaror däremot också är försedda med något slags testprotokoll som på ett trovärdigt sätt visar att de uppfyller vissa grundkrav vad det gäller funktion och kvalitet, då skulle dessa programvaror vara av särskilt intresse för den presumtive köparen. Denne skulle redan före köpet få vissa garantier för programvarans funktion och kvalitet, utan att själv behöva utföra några tester.

Om svensk programvaruindustri engagerar sig för ett sådant testförfarande skulle både programvaruföretag och svensk programvaruindustri som helhet stärkas.

Därmed har vi åtminstone fyra kategorier som har intresse av på något sätt mäta och utvärdera en programvaras kvalitet: slutanvändaren, marknadsföraren, utvecklaren av programvarorna samt programvaruindustrin.

Det är inte så svårt att hitta ytterligare kategorier, men för det fortsatta resonemanget kan vi nöja oss med fyra. De representerar delvis olika intressen och har olika uppgifter att utföra. Frågan är om de också kan nå fram till samma uppfattning om vad som menas med kvalitet?

Hur ska utvecklaren av en programvara kunna ge den en kvalitet som marknadsföraren kan använda i sin marknadsföring och som samtidigt är den kvalitet som köparen och användaren förväntar sig att den inköpta produkt har?

Gemensamt kvalitetsbegrepp

Det är ju uppenbart att de olika intressenterna, antingen de är fyra eller ännu flera, på något sätt måste enas om vad man egentligen menar med kvalitet. Vi måste med andra ord hitta definitioner och mätmetoder med vars hjälp man kan mäta en programvaras kvalitet på ett så objektivt sätt som möjligt. De resultat man får fram genom dessa mätningar ska ge en bild av programvarans kvalitet som varje intressent kan acceptera som korrekt.

För att mätningen ska vara trovärdig för de olika parterna bör mätmetoderna kunna redovisas på ett öppet sätt och verkligen mäta den kvalitet man vill mäta.

Problemet är på sätt och vis likartat det som gäller vid mätning av kvantitet. Den som köper ett kilo potatis i lös vikt vill vara säker på att han verkligen får ett kilo och inte mindre. Därför väger han sina potatisar på kundvågen och ser exakt vilken vikt påsen med sitt innehåll har. Men hur vet han att kundvågen, som står under säljarens tak, verkligen väger rätt och inte till säljarens fördel? Svaret är att kundvågen är kontrollerad av en oberoende provningsanstalt, som därför försett vågen med ett kontrollmärke som garanterar att vågen håller måttet.

Kontrollmärkningen av vågar garanterar således att såväl säljare som köpare kan lita på att det som vägs på vågen har den vikt som vågen anger. Därigenom kan parterna lugnt ägna sig åt själva affärerna, vilket är vad de helst vill hålla på med.

Men om det nu inte finns något kontrollorgan? Hur gör man då?

Öppenhet ger trovärdighet

Låt oss för ett ögonblick dröja oss kvar vid vägningsproblemet och lära något av historien. Från år 1250 finns ett av de första dokumenterade fallen av falskvägning. Då rapporterades nämligen om en utrustning med vars hjälp man avsiktligt kunde väga till den ena partens fördel. Utrustningen bestod av en balansvåg (förmodligen en besman) med ihålig bom som innehöll kvicksilver. Kviksilveret fungerade som osynlig ballast och höll vågen i balans vid felaktig vikt. Den förfördelade hade ingen möjlighet att kontrollera detta.

Om däremot balansvågen hade varit transparent så skulle båda parter haft möjlighet att kontrollera vågens konstruktion och funktion och då hade det knappast gått att fuska.

Lärdomen av detta exempel är att öppenhet i affärer är väsentligt, både för trovärdigheten och möjligheten för de olika parterna att kontrollera varandras uppgifter. Om den part som säljer en vara öppet kan redogöra för och visa hur han vägt eller mätt den aktuella produkten, och den tilltänkte köparen samtidigt själv kan kontrollera dessa uppgifter, då lägger man en bra grund för produkter med bra kvalitet och därmed bra affärer.

Öppenhet är väsentlig och det gäller antingen man talar om kvantitet eller kvalitet.

Om man således kan få fram en metod med vars hjälp ett företag kan värdera sin programvaras funktion och kvalitet och andra intressenter kan

kontrollera denna värdering, då har man uppnått något som bör vara av intresse för många parter.

Kvalitetsmärkning ger många fördelar

Utvecklaren skulle då få ett instrument med vars hjälp han eller hon redan under utvecklingen kunde testa, förbättra och bygga in kvalitet i sin produkt.

Säljaren kunde gentemot köparen garantera att programvaran uppfyller vissa villkor och får därigenom ett bra försäljningsargument.

Köparen skulle lättare kunna välja programvara och känna större trygghet i köpet. Själva köpet underlättas också.

Slutanvändaren, slutligen, skulle få garanti för att programmet fungerar bättre, vilket leder till mindre avbrott. Därigenom kan han eller hon bättre ägna sig åt sitt egentliga arbete, där användningen av programvaran stödjer hans verksamhet utan att störa det genom avbrott eller liknande. Användarens arbetsmiljö förbättras samtidigt som arbetet effektiviseras.

Vi ser således att det inte bara finns ett antal intressenter som bör ha intresse av kvalitetsmärkta programvaror. Själva kvalitetsmärkningen ger en mängd fördelar som bl a kan påverka såväl arbetsmiljö som effektivitet.

Det finns givetvis många önskingar och behov att tillgodose. Man ska inte räkna med att allt kan uppfylla på en gång. Utvärdering av programvaruprodukter ska ses som en process, där det första steget inte är det sista och kanske inte heller det bästa, men kanske det viktigast och svårast. Om man vågar ta det första steget och samtidigt har en vision att gå vidare så har man kommit en bit på väg. Även en tusenmilsresa börjar med ett första steg.

Ett första steg kan bestå i att programvaruföretaget under ordnade former, öppenhet och med internationella standarder som grund självt genomför utvärdering av sin programvara. Utvärderingen eller egendeklarationen garanteras av Svensk programvaruindustri, SPI, som utfärdar certifieringsintyget.

Denna utvärderingsmetod är redan en realitet och har provats på ett begränsat antal programvaror. Tiden är nu mogen att sprida kunskap om denna metod och förmå programvaruföretagen att anamma den möjlighet som nu erbjuds. Egendeklaration av programvaror kan bli ett första, kraftfullt steg för företagen att förbättra sina programvarors funktion och kvalitet och öka sin egen konkurrenskraft, bl a genom att kunderna kan förväntas bli nöjdare.

Egendeklaration – ett första steg

På sikt kan man tänka sig att fristående institut genomför utvärdering och certifiering av programvaror. En intressant förebild skulle kunna vara den testning och märkning av bildskärmar och datorer som TCO byggt upp och som nu är väl känd, inte bara i Sverige, utan över hela världen. Några rader om hur detta började och sedan har utvecklats kan utgöra inspiration för det fortsatta arbetet med utvärdering och certifiering av programvaror.

Det var 1990 som TCO inledde ett samarbete med Naturskyddsföreningen, vilket så småningom resulterade i skriften ”Det miljövänliga kontoret”. Senare bildade TCO en utvecklingsenhet och började samarbeta med NUTEK:s kansli för effektivare energianvändning samt SEMKO AB. Arbetet ledde 1992 fram till en metod för miljömärkning av bildskärmar, vilken fick beteckningen TCO '92.

Ett par år senare hade ett 20-tal internationella data- och bildskärmsföretag ansökt om och fått TCO-certifiering av ett 60-tal bildskärmar.

Märke som ger exportintäkter

Intresset var så stort att TCO i samarbete med Exportrådet på våren 1994 kunde etablera ett TCO Information Center i Chicago för att informera om miljömärkning av bildskärmar. Detta har efterhand gett exportintäkter från certifiering av bildskärmar och försäljning av skrifter och ”provning-verktyg”. För 1997 uppgick intäkterna i form av certifieringsavgifterna till 20 miljoner kronor. Huvuddelen av dessa var exportinkomster.

TCO '92 har efter hand utvecklats och lett fram till TCO '95, där man breddat kraven till att omfatta såväl ekologiska som arbetsmiljömässiga krav liksom krav på energieffektivitet för bildskärm, systemenhet och inmatningsdon. Detta har i sin tur krävt att man till projektet knutit experter från andra områden, bl a inom ergonomi, emission, ekologi och effektiv energianvändning.

Totalt har ett hundratal tillverkare och leverantörer runt om i världen anslutit sig till TCO-standarden och i september 1998 var mer än 1 700 modeller certifierade.

TCO '92 – Sveriges mest spridda varumärke

Idag är det gröna TCO '92-märket det mest spridda svenska varumärket i

världen! Det har blivit en väl etablerad de facto standard, på sätt och vis mot alla odds. I varje fall om man betänker att den svenska marknaden för bildskärmar bara utgör en enda procent av världsmarknaden. Ändå blev det möjligt att med denna lilla procent som grund sätta en standard för övriga världen.

Detta bör kunna inspirera arbetet med att också etablera en standard för utvärdering och certifiering av programvaror. En hel värld skulle ha nytta av detta.

SPI har nu presentera det första genomarbetade systemet i världen som med utgångspunkt från ISO/IEC-standarder ger företag möjlighet att egen-deklarera programvaror mot fördefinierade kravnivåer. Dessa egendeklarationer blir tillgängliga via Internet, där såväl testresultaten som de metoder som har använts vid mätning och utvärdering av de olika programvarorna.

Följande sidor berättar om grunden för denna metod, hur den tillämpas och vilka erfarenheter som i dag finns från användningen av metoden.

2. ISO-standarder som grund för utvärdering av programvaror

I föregående kapitel har vi berört olika aspekter på kvalitet liksom att det finns många människor och intressenter som skulle ha nytta av om man på ett objektivt sätt kunde utvärdera och få ett mått på en programvaras kvalitet med avseende på funktion m m. Det förutsätter att de olika intressegrupperna kan enas om vad man menar med kvalitet. Är det möjligt att nå fram till en sådan samsyn? Frågan är berättigad eftersom de berörda parterna har delvis olika behov. Än större kan problemet tyckas vara om man tänker på att programvaror har så olika funktioner och kan utföra så varierande uppgifter i de mest skilda sammanhang.

Ändå är svaret på frågan om man kan nå fram till en samsyn ”ja”. Det förutsätter dock två saker: dels att vi enas om en definition för just mjukvarukvalitet, dels att vi skapar en kvalitetsmodell som tydliggör kvalitetsbegreppets olika aspekter.

Mjukvarukvalitet

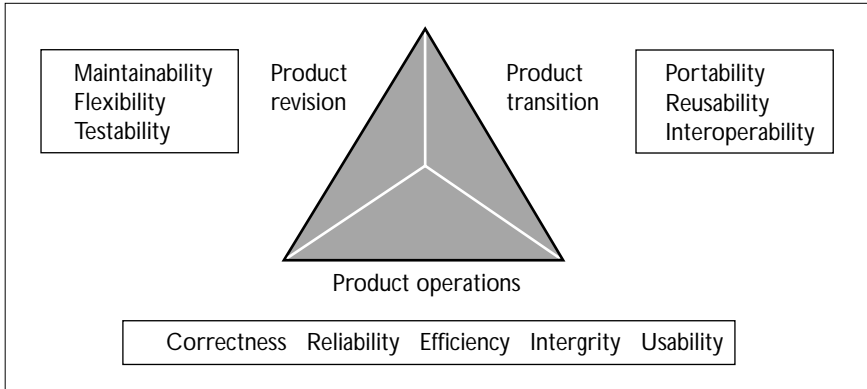
Vad det gäller mjukvarukvalitet så har man på internationell standardiseringsnivå, ISO/IEC 8402, enats om en definition enligt följande:

”The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs”.

Detta ska tolkas så att kvalitet i detta sammanhang inte bara kan definieras som en mjukvara utan fel. Mjukvarans kvalitet måste beskrivas mer detaljerat än så. Det blir möjligt att göra om man använder en kvalitetsmodell, där kvalitetsbegreppet tydliggörs.

Kvalitetsmodeller har diskuterats under lång tid. Ett exempel på tidig kvalitetsmodell föreslogs redan 1977 av McCall, som då arbetade på US Air force vid RADC (Rome Air Development Center), vilket nu kallas

Rome Laboratories. McCall och hans kolleger föreslog då följande modell:



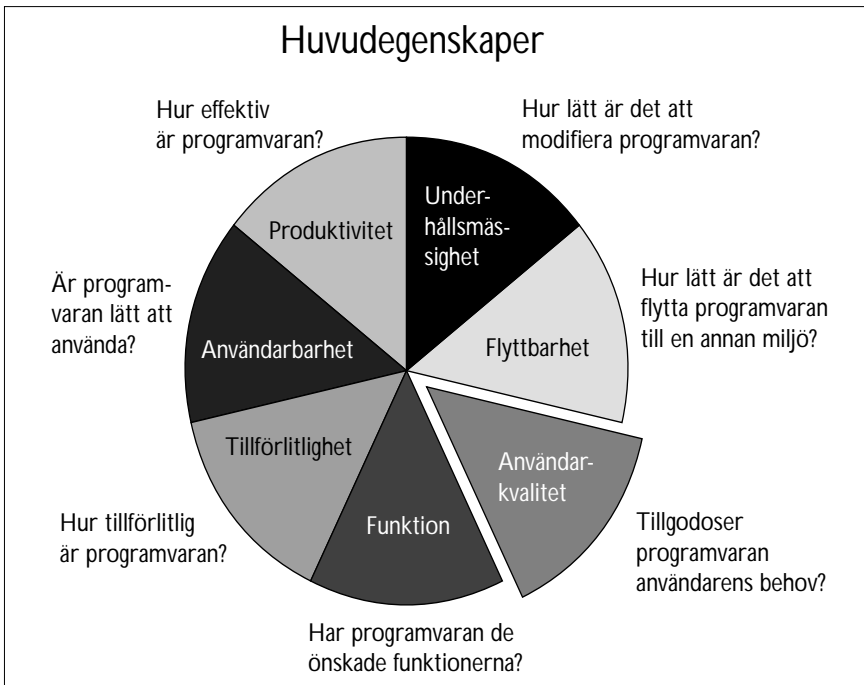
Kvalitetsmodell från 1977 av McCall

Modellen är baserad på tre typer av kvaliteter:

- extern kvalitet som den uppfattas och beskrivs av användaren
- intern kvalitet som den uppfattas och beskrivs av utvecklaren
- kvalitet (för kontroll i form av en kvantitativ skala och metod som kan används för mätning (ISO/IEC 14598-1))

Flera andra modeller har efter hand tagits fram och behovet att enas kring en kvalitetsmodell för mjukvaror har så småningom resulterat i en standard. Från början utgick man från sex huvudegenskaper. Så småningom tillkom ytterligare en. Totalt finns det alltså sju huvudegenskaper. Varje huvudegenskap kan sägas ge svar på en eller flera frågor.

Huvudegenskaperna kan på ett övergripande sätt illustreras enligt följande.



Från början utgick man från sex huvudegenskaper. Den sjunde, användarkvalitet, tillkom senare.

De sex huvudegenskaperna

Om vi börjar med de sex huvudegenskaperna kan de i svensk översättning definieras enligt följande. (Det bör påpekas att de svenska översättningarna är gjorda i ett tidigt skede och stämmer inte helt överens med de nyare, engelska motsvarigheterna som efter hand korrigerats något. Ändå ger de svenska översättningarna en någorlunda bra bild av vad ISO menar med de sex huvudegenskaperna.)

Funktion (Funktionalitet): en mängd av attribut som har sin grund i en uppsättning funktioner och deras specificerade egenskaper. Funktionerna är de som tillgodoser uttalade eller underförstådda behov. [ISO/IEC 9126: 1991]

Tillförlitlighet: en mängd av attribut som gör att programvaran kan upprätthålla sin prestandanivå under givna villkor för en given tidsperiod. [ISO/IEC 9126: 1991]

Användbarhet: en mängd av attribut som avgör den prestation som är

nödvändig för användning, och på den enskilda utvärderingen av en sådan användning, av givna eller en underförstådd grupp användare. [ISO/IEC 9126: 1991]

Produktivitet (effektivitet): en mängd av attribut som beror på förhållandet mellan det som programvaran utför och mängden av resurser som under givna villkor använts. [ISO/IEC 9126: 1991]

Underhållsmässighet: en mängd av attribut som avgör vilka resurser som är nödvändiga för att utföra specificerade ändringar. [ISO/IEC 9126: 1991]

Flyttbarhet: en mängd av attribut som avgör vilka resurser som krävs för att flytta programvaran från en miljö till en annan. [ISO/IEC 9126: 1991]

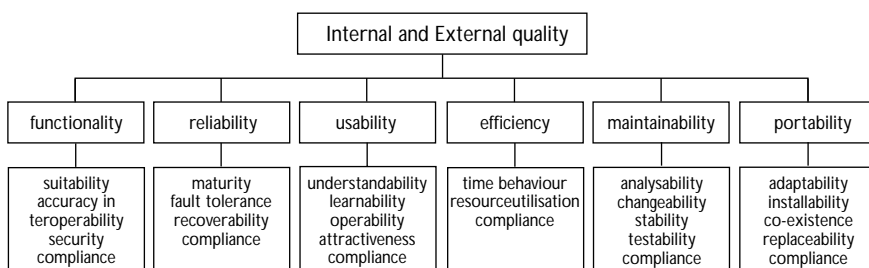
De sex huvudegenskaperna definierar kvalitetsbegreppet på hög nivå. För att kunna gå vidare och även kunna göra mätningar måste vi mer i detalj kunna definiera varje huvudegenskap. Detta görs med hjälp av underegenskaper.

Underegenskaper och attribut

Underegenskaperna manifesterar sig när programvaran används som en del av ett datorsystem och är resultat av interna mjukvaruattribut.

Ett attribut kan beskrivas som en väldefinierad del eller funktion av produkten och kan vara intern eller extern. Definition av attribut spelar en avgörande roll vid utvärdering av programvaror, vilket vi kommer att beröra mera i ett senare kapitel.

Om vi således utgår från de sex ursprungliga huvudegenskaperna och bryter ner dessa till underegenskaper, får vi följande bild hämtad från ISO/IEC 9126:



Intern och extern kvalitet med sex huvudegenskaper och 27 underegenskaper enligt ISO 9126-1. Bilden återges med tillstånd från ISO/IEC.

Användarkvalitet

Från början tänkte man sig att dessa definierade huvudegenskaper och underensskaper skulle vara tillräckliga för att ge en komplett bild av mjukvarukvalitet. Emellertid visade det sig att det behövdes ännu ett begrepp för att göra kvalitetsbegreppet fullödigt. Vad som krävs anar man när man läser den definition av kvalitet som görs i ISO8402:1994. Där står att kvalitet är "alla sammantagna egenskaper hos ett objekt eller företeelse som ger dess förmåga att tillfredsställa uttalade eller underförstådda behov".

Det sistnämnda ordet "behov" syftar på användarens behov när denne använder sin programvara. Vi behöver därför införa ytterligare ett begrepp för att få en komplett bild av programvarukvalitet, nämligen Quality in use, dvs användarkvalitet. Med detta uttryck beskriver man programvarans kvalitet utifrån användarens synpunkter, när denne använder sin programvara.

Quality in use eller Användarkvalitet definieras med hjälp av fyra underensskaper, där de engelska orden står inom parentes: effektivitet (effectiveness), produktivitet (productivity), säkerhet (safety) och tillfredsställelse (satisfaction)

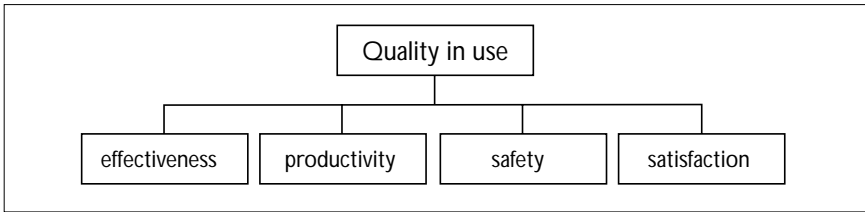
För att förhindra missförstånd kan det här vara på sin plats att kort beröra de engelska begreppen "effectiveness" och "productivity".

Med "effectiveness" menas hur mjukvaran ger användaren möjlighet att uppnå specificerade mål. Det har inget att göra med resurser, utan avser istället hur väl man planerar sina resurser så att man når ett visst resultat mot det man planerat. Begreppet är resursoberoende.

"Productivity", däremot, syftar på hur användaren kan utnyttja mjukvaran för att utifrån vissa tillförda resurser med viss effektivitet nå vissa resultat, dvs det beskriver förhållandet mellan de resurser man sätter in och det resultat man når. Begreppet är resursberoende. Med resurser avses exempelvis tid, material, kostnader.

De övriga två begreppen som krävs för att definiera användarkvalitet vållar nog mindre problem. "Säkerhet" eller "safety" syftar på den nivå programvaran har med avseende på risken att orsaka skada på människor, mjukvara, utrustning eller omgivning i ett specificerat sammanhang. "Tillfredsställelse" eller "satisfaction" avser mjukvarans förmåga att tillfredsställa användaren i ett specificerat sammanhang.

Med detta som grund kan vi teckna följande bild av användarkvalitet.



Kvalitet vid användning innefattar en huvudegenskap och fyra underegenskaper enligt ISO 9126-1. Bilden återges med tillstånd från ISO/IEC.

Tre steg till kvalitet

Vi har nu definierat intern och extern kvalitet och därtill användarkvalitet bestående av sammanlagt sju huvudegenskaper, vilka kan brytas ner i 31 underegenskaper. Dessa underegenskaper representeras av noggrant fastställda attribut och det är dessa man mäter med hjälp av kvantitativa metoder. Detta kan sammanfattas i följande trestegstrappa, som utgör den teoretiska grunden för hur man bedömer en programvaras kvalitet:

- *Egenskaper* i form av sex huvudegenskaper; funktion, pålitlighet, användbarhet, effektivitet, underhållbarhet, flyttbarhet. Till detta lägger vi dessutom användarkvalitet som en särskild egenskap. Vi har därmed sju huvudegenskaper som i sin tur kan delas in i:
 - *Underegenskaper*, 31 st, vilka kan värderas och mätas med hjälp av:
 - *Metrik*¹; en kvantitativ skala och metod som kan användas för mätning av underegenskaper (ISO 14598)

De 31 underegenskaperna är på engelska:

Functionality: suitability, accuracy, interoperability, security, compliance

Reliability: maturity, fault tolerance, recoverability, compliance

Usability: understandability, learnability, operability, attractiveness, compliance

¹ Ordet metrik, kommer från latin *metrica*, av *metricus*, "hörande till mått" och med liknande betydelse i grekiska *metrikos*, av *metron*, "mått", "verktyg att mäta med". På svenska används metrik vanligtvis om versmått i poesi. Samma sak gäller på engelska, men där har ordet "metrics" också betydelsen "the theory or a system of measurement". Den sistnämnda betydelsen ligger närmast den betydelse som "metrik" har här.

Efficiency: time behaviour, resource utilisation, compliance

Maintainability: analysability, changeability, stability, testability, compliance

Portability; adaptability: installability, co-existence, replaceability, compliance

Quality in use: Effectiveness, productivity, safety, satisfaction

Utvärdering av en mjukvaruprodukts kvalitet görs genom att man mäter interna attribut (typically static measures of intermediate products) och genom att mäta externa attribut (typically by measuring the behaviour of the code when executed).

Som framgår av det som hittills diskuterats så har det varit nödvändigt att definiera programvarukvalitet både med hjälp av olika begrepp och med olika utgångspunkter, t ex intern och extern kvalitet samt användarkvalitet.

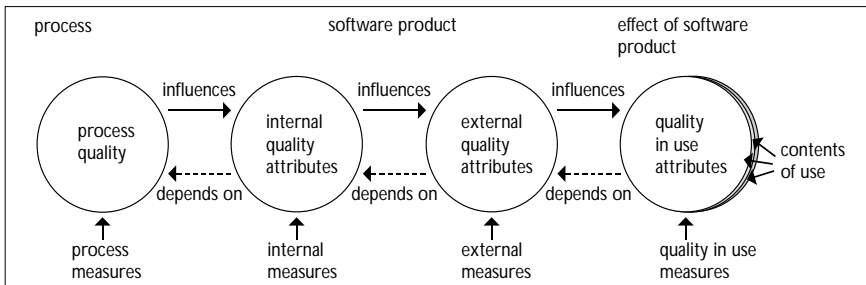
Livscykelkvalitet

Det som har sagts ovan skulle kunna utgöra tillräckligt underlag när vi fortsättningsvis diskuterar hur själva utvärderingen kan göras. Man kan emellertid vidga själva kvalitetsbegreppet ytterligare till att också omfatta kvaliteten på den process inklusive det arbete som leder fram till själva programvaran, dvs processkvalitet. Denna process styr i hög grad programvarans slutliga kvalitet och är därför av stort intresse, eftersom man även under utveckling av programvaran kan göra utvärderingar för att i ett tidigt skede få ett mått på själva programvarans kvalitet.

Processkvalitet bidrar till produktkvalitet och produktkvalitet bidrar i sin tur till användarkvalitet. Genom att förbättra processkvaliteten, förbättrar man således produktkvaliteten och ytterst användarkvaliteten.

På liknande sätt kan man säga att om man utvärderar användarkvaliteten får man en återkoppling som gör det möjligt att förbättra produktkvaliteten. Likaså kan utvärdering av programvaran som produkt ge en återkoppling som gör det möjligt att förbättra processkvaliteten.

Däriigenom får man ett samband i hela den livscykel som omfattar en programvarans tillkomst, utveckling och livslängd. Dessa olika kvalitetssamband kan illustreras med nedanstående bild hämtad ur ISO/FDIS 9126-1.



Kvalitet i livscykel enligt ISO/IEC 9126-1. Bilden återges med tillstånd från ISO/IEC.

Att genomföra utvärderingen

De samband som här har visats vad det gäller programvarans totala kvalitet och framför allt den kvalitetsmodell som har beskrivits, utgör grunden för den tillämpning och metod som också måste till för att handgripligen utvärdera programvaror. Andra väsentliga delar för utvärderingen är den utvärderingsmetod man väljer, själva mätningarna samt de verktyg som används under utvärderingen.

Avsikten med utvärdering av programvaror är ju att få fram kvantitativa resultat som rör mjukvarans kvalitet. Resultaten måste dessutom vara begripliga, tillförlitliga och kunna accepteras av alla intressenter.

Själva kvaliteten beskrivs med hjälp av egenskaper och underegenskaper. De sistnämnda begreppen styrs av produktattribut, vilka i sin tur utgör grunden för de mätningar som genomförs under utvärderingens gång.

Produktattribut kan bestå av begrepp som exempelvis: söka artikel, makulera kvitto, skriva ut etc. Se vidare exempel på nästa sida som visar en huvudegenskap, dess fem underegenskaper och några produktattribut. Ett produktattribut ska vara en mätbar egenskap i produkten. Ett attribut kan vara kvantitativt, t ex storlek av något slag, eller kvalitativt, då man exempelvis gör en värdering på en viss skala.

Går man till själva utvärderingsprocessen så startar den i princip när någon begär att en programvara ska utvärderas. Den som begär utvärdering kan exempelvis vara:

- programvaruutvecklare
- programvaruleverantör
- programvaruköpare
- programvaruanvändare
- programvarutillverkare (i sin roll som programvaruköpare)

Egenskaper	Underenskaper	Grundkrav	Verifikat	Produktattribut
1 Functionality	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.1	Starta/Stäng kassa (logga in)
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.2	Enkel försäljning med en artikel (även avbryta)
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.3	Öppna kassalåda
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.4	Makulera kvitto
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.5	Parkera kvitto
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.6	Sök artikel
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.7	Konfigurering/Editera kvitto
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.8	Specialkvitton (reklamation, retur)
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.9	Rabatter
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.10	Hjälp
	1.3 Interoperability	1.3.1 hantera basutrustning och gränssnitt	1.3.1.2	stödja kringutrustning för kassa
	1.4 Security	1.4.1 grundläggande datasäkerhet	1.4.1.2	skyddad kvittojournal
1.5 Compliance	1.5.1 stöd för branschstandarder	1.5.1.3	god redovisningssed	

Grundkravet "funktion" med några av dess fem underenskaper och hur dessa kan beskrivas i form av grundkrav och motsvarande produktattribut. (Verifikat refererar till mätmetoder m m).

En utvärderare kan t ex vara:

- tredje parts provningslaboratorium
- provningsenheter inom programvarutillverkning eller distributionsorganisationer

- provningsenheter inom programvaruköpare eller användarorganisationer
- provningsenheter inom systemtillverkningsorganisationer
- organisationer som utför jämförelser mellan produkter

De båda parterna – uppdragsgivare och utvärderare – träffar en överenskommelse om att en utvärdering ska göras. Man kommer också överens om de specifikationer som ska gälla för utvärderingen.

Innan utvärderingen tar sin början ska uppdragsgivaren lämna en preliminär version av utvärderingskraven. Vidare ska han göra en beskrivning av den programvara som ska utvärderas inklusive de komponenter som hör till. I detta kan exempelvis ingå konstruktionsdokument, utvärderings- eller valideringsrapporter, källkoder och annan användardokumentation.

Utvärderaren ska tillhandahålla fördefinierade utvärderingsspecifikationer, utvärderingsmetoder och utvärderingsverktyg.

Utvärderingsmetoden beskrivs i ISO 14598, som för närvarande består av fem delar. I det här sammanhanget är del fem av störst intresse, eftersom den beskriver krav och rekommendationer för den praktiska implementeringen vid värdering av mjukvaruprodukter. Här definieras de aktiviteter som behövs för att analysera värderingskraven, liksom specificering, utformning och genomföring av värderingsåtgärderna samt de slutsatser som kan dras efter utvärderingen.

Utvärderingsprocessen innehåller ett antal aktiviteter som utförs i samarbete med den som begärt utvärderingen. Utvärderingen bygger på data från den som begärt utvärderingen, data från utvärderaren och sådant som kommit fram från andra aktiviteter.

Utvärderingskrav

Själva utvärderingsaktiviteterna ska uppfylla vissa krav. Bl a ska mätningarna eller motsvarande aktiviteter vara:

- repeterbara; upprepad utvärdering av samma produkt för samma utvärderingsspecifikation av samma utvärderare ska ge samma resultat som den första utvärderingen
- reproducerbara; utvärdering av samma produkt för samma utvärderingsspecifikation av en annan utvärderare ska ge samma resultat som från den första utvärderaren

- opartiska; utvärderingen är inte riktad mot något speciellt resultat
- objektiva; utvärderingsresultaten är grundade på aktuella fakta, ofärgade av utvärderarens känslor eller åsikter

Däremot kan utvärderingar av samma produkt genomföras med olika utvärderingsspecifikationer, beroende på vilket mål man satt. Om man således gör två utvärderingar av samma programvaror men med olika mål kan resultaten bli olika och är därför inte jämförbara.

Utvärdering i fem steg

Själva utvärderingsprocessen delas in i fem steg:

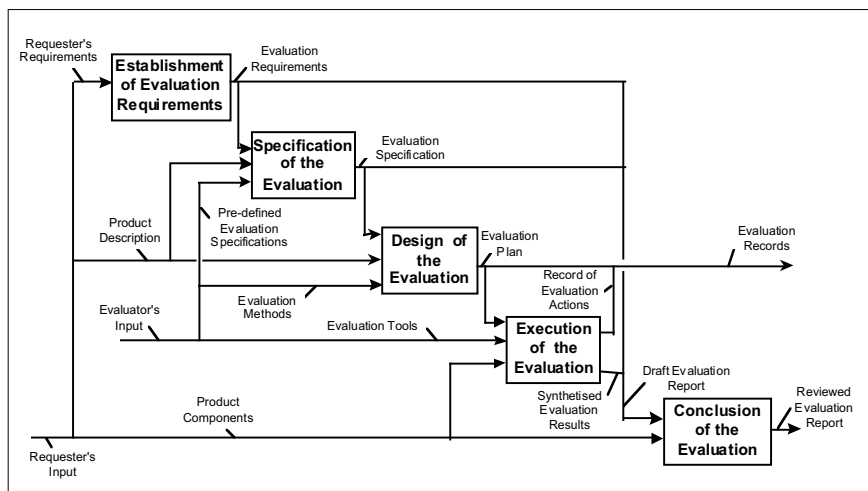
- fastställande av utvärderingskraven
- specificering av värderingen baserad på ovannämnda, fastställda värderingskrav. I detta ingår också den beskrivning av produkten som beställaren av utvärderingen lämnar.
- utformning av värderingen i form av en utvärderingsplan baserad på specifikationen. I denna tar man hänsyn till programvaran och de komponenter (delar) som ingår, liksom den utvärderingsmetod som utvärderaren föreslår.
- genomförande av utvärderingsplanen. I denna kan ingå inspektion av programvaran, modellering, mätning och testning av produkten och dess olika komponenter i enlighet med planen. Dessa aktiviteter kan utföras med olika programvaruverktyg. Aktiviteterna redovisas och resultaten skrivs ner i ett preliminärt protokoll.
- sammanfattning av utvärderingen som innefattar leverans av utvärderingsrapporten. Dessutom ska utvärderaren förstöra utvärderade produktkomponenter.

Utvärderingsrapport

Utvärderingen ska leda fram till följande dokument:

- utvärderingsdata, utvärderingsplan samt övriga data från själva utvärderingen
- utvärderingsrapport, som också ska innehålla värderingskraven, värderingsspecifikation samt sammanställning av utvärderingsresultaten
- granskad och slutgiltig utvärderingsrapport

Själva utvärderingsprocessen kan schematiskt beskrivas med hjälp av nedanstående figur.



Schematisk beskrivning av utvärderingsprocessen. Bild hämtad från ISO 14598-5. Bilden återges med tillstånd från ISO/IEC.

Kostnader

I första hand kan utvärderingsmetoden användas för programvaror som inte är alltför omfattande. Det är i och för sig inget som hindrar att man med samma metod även testar stora programvaruprodukter. Ett sådant arbete kommer emellertid att bli ganska resurskrävande och blir därför dyrbart att genomföra.

För mindre omfattande programvaror bör man räkna med att en utvärdering inte bör kosta mer än i mellan 30 000 – 40 000 kronor.

Därtill kommer en registreringsavgift på 5 000 kronor samt en årsavgift. Årsavgiften är tänkt att utgöra 0,1 promille av produktens årliga försäljningsvärde med ett minimumbelopp av 5 000 kronor och ett maximumbelopp på 25 000 kronor.

Objektivitet och jämförbarhet

När vi därmed även har de ekonomiska ramarna klara kan det vara dags att gå från teori till praktisk handling genom att studera hur en utvärdering i

verkligheten kan gå till. Vi gör det dock med utgångspunkt från en sammanfattning av metodens centrala egenskaper. Dessa är:

- **Objektivitet**, genom kravet på användning av etablerade accepterade och objektiva metoder för kvalitet enligt ISO-standarderna 9126-1 och 14598-5 för att identifiera och fokusera arbetet på de mest väsentliga egenskaperna
- **Räckvidd**, genom att vara icke-selektiva, dvs tillämpbara för alla programprodukter på marknaden inom definierade produktgrupper
- **Neutralitet**, genom frånvaron av värderingar och förutbestämda kravnivåer som skall vara uppfyllda
- **Öppenhet**, genom lättillgänglig information på Internet
- **Kompetenshöjande**, genom att förklaringar av definitioner och begrepp samt allmän information om kvalitet länkas till uppgifterna om egendeklarationer på Internet
- **Kostnadseffektivitet**, genom att egendeklarationer bygger på redan befintliga arbetssätt och öppna etablerade system
- **Flexibilitet**, genom att innehållet i en egendeklaration kan ändras när så är påkallat och så önskas ske av programvaruföretaget efter vederbörlig intern granskning och kontroll
- **Jämförbarhet**, genom kravet på utarbetande av produktspecifika regler som skall följas vid framtagning av underlagsmaterial och presentation av egenskaper i egendeklarationen.

3. Egendeklaration bygger på regler och öppenhet

Hur går man då tillväga vid utvärdering av en programvara enligt de metoder som finns? Hur bestämmer man vad som ska testas och mätas? Och till vem vänder man sig?

Till att börja med kan man konstatera att det ännu inte finns något fristående institut som kan utföra utvärderingen enligt de riktlinjer som vi tidigare diskuterat. Metoden är ju i sin tillämpning ny och håller på att sjösättas. Ändå har några företag redan fått sina programvaror utvärderade. Hur är detta möjligt?

Svaret är att i brist på fristående utvärderingsinstitut eller kontrollanstalter får man tillgripa egendeklarationer. Det är ingen nödlösning utan en praktisk, framkomlig väg som gör att man kan komma igång. Egendeklaration innebär att företaget självt genomför tester, i de aktuella fallen med hjälp av viss extern expertis. Har företaget egen, tillräcklig kunskap att genomföra egendeklaration kan testerna göras utan externa experter – under förutsättning att man följer de regler som de internationella standarderna förespråkar.

Den utvärderingsmetod som nu börjat användas har tillkommit genom mångårigt arbete. SPI tog itu med utvärderingsfrågor av programvaror redan i början av 1990-talet. Under senare år har föreningen koncentrerat sig på att ta fram ett system för utvärdering av programvaror genom egendeklaration. Systemet har utvärderats i samarbete med mjukvarubranschen och inte minst med SPI:s egna medlemsföretag.

Objektivt och trovärdigt

I de regler som nu utarbetats av SPI står det att: ”Egendeklarationer skall ge en möjlighet att objektivt och trovärdigt kunna ge beskrivning av en programvaras kvalitetsegenskaper. Primärt avses egendeklarationer kunna användas av professionella inköpare inom näringsliv och förvaltning.”

Den sista meningen syftar på att inköpare kan använda egendeklarationerna för att bättre kunna bedöma de programvaror man har för avsikt att köpa. Det blir lättare att se vad en programvara verkligen går för innan man bestämmer sig. På lite sikt kan man också tänka sig att inköparna ställer krav på att de programvaruföretag som vill ge anbud vid en upphandling måste egendeklarera sina programvaror för att de överhuvudtaget ska beaktas.

Vidare heter det att: ”Grundsytet med egendeklarationer är att de skall vara tillämpbara för alla programvaror samt beskriva utvalda kvalitetsegenskaper. Egendeklarationer är en öppen redovisning av kvalitetsegenskaper och är neutrala i och med att inga förutbestämda kravnivåer finns angivna.”

Egendeklarationen av produktens kvalitetsegenskaper innehåller ingen värdering som direkt kan åskådliggöras t ex på en tiogradig skala eller liknande. Den som läser en egendeklaration får göra sina ställningstaganden och bedömningar utifrån egna utgångspunkter och baserat på de uppgifter om produktens kvalitetsegenskaper som anges i egendeklarationen.

En tilltänkt kund som studerar egendeklarationen får ett mått på programvarans kvalitet och funktion *innan* han köper varan. Detta är naturligtvis av mycket stort värde. För leverantören eller för den som tagit fram programvaran innebär utvärderingen en möjlighet att kvalitetssäkra sin produkt och efter hand höja kvaliteten ytterligare.

Tanken är att exempelvis ett programvaruföretag genom egendeklaration ska få en möjlighet att genomföra utvärdering av sin programvara och sedan öppet kunna redovisa såväl utvärderingsmetod som mätdata, resultat och slutsatser. Egendeklaration är ett första steg för programvaruleverantörer att kunna lämna en genomarbetad varudeklaration på sina egna produkter.

Öppenhet avslöjar fusk

Innan vi gå vidare kan det vara rimligt att något skärskåda eventuella svagheter som några läsare kanske ser i metoden med egendeklarationer.

Som ett litet kuriosum kan man kanske först säga att egendeklaration också skulle kunna kallas för självdeklaration. Det leder tankarna till de blanketter som vi varje år måste lämna till myndigheterna där vi ska uppge våra inkomster enligt vissa regler. Även om inte alla uppgifter vi lämnar är offentliga för alla människor så har vi även här en hög grad av öppenhet och ärlighet som viktiga inslag för att systemet ska fungera. Det finns dess-

utom möjlighet till sanktioner mot dem som visat sig bryta mot gällande bestämmelser och inte varit ärliga. Säg vad man vill om våra självdeklARATIONER, de fungerar på det hela taget ganska bra och utgör en grundbult i skattesystemet.

Egendeklarationer av programvaror förutsätter på liknande sätt att utvärderaren är ärlig. Eftersom öppenheten här är större så kan man nog på goda grunder räkna med bra resultat. Visserligen kan man naturligtvis tänka sig att en fuskande utvärderare skulle kunna ta fram en felaktig eller förskönad deklARATION av programvaran. Det skulle möjligen kortsiktigt kunna ge vissa fördelar, men bara kortsiktigt. Hela utvärderingsförfarandet måste ju nämligen redovisas öppet, inklusive de metoder och mätningar som gjorts. När fusket väl uppdragas skulle det snabbt få förödande konsekvenser för den som fuskar.

Just därför att systemet är öppet blir det möjligt för var och en att kontrollera praktiskt taget allt som gjorts. Och upptäcker en kund eller konkurrent att ett företag fuskat eller slarvat med sin egendeklaration, så är inte bara den produkt som fått en felaktig deklARATION snabbt död på marknaden. Ett avslöjat fusk lär ju drabba hela företaget, som fortsättningsvis skulle ha små möjligheter att marknadsföra även andra programvaror, i varje fall under sina gamla företagsnamn eller produktnamn.

Konkurrenterna skärper kraven

En annan invändning skulle kunna vara att eftersom utvärderaren själv sätter kravnivån för utvärderingen, skulle denna kunna sättas så låg att testresultatet visade ett mycket bra betyg med många godkända delprov och kanske inget underkänt. Detta är i sig inget fusk, men det är ändå ett tämligen kortsiktigt och ogenomtänkt agerande om man betänker att informationen är öppen. I synnerhet konkurrenterna skulle snabbt upptäcka den låga kravnivån och de kan då välja en högra nivå på utvärderingen av sina egna programvaror. Dessa kommer då med rätta att framstå som intressantare för kunderna och skulle antagligen tvinga det första företaget att göra en ny urvärdering av sin programvara med högre krav – förmodligen sedan man först gjort avsevärda förbättringar på den.

Man kan på goda grunder räkna med att just företagen själva kommer att hålla sig underrättade om vilka kravnivåer konkurrenterna valt på sina testade programvaror. Att en sådan förväntan är riktig kan belysas med exemplet när SEMKO:s myndighetsroll förändrades och företagen själva

fick ansvar för att leva upp till det s k lågspänningsdirektivet. Då var det just de konkurrerande företagen själva som slog larm när de upptäckte att det fanns andra företag som inte levde upp till direktivet.

Det är på liknande sätt som utvärderingsmetoden av programvaror förhoppningsvis kommer att fungera på lite sikt. Branschen sätter själv nivån på utvärderingen, men konkurrensen och öppenheten kommer att tvinga fram allt högre, dokumenterade kvalitetskrav.

Så långt är vi ännu inte, men vi är tillräckligt långt för att kunna hävda att systemet verkligen kan fungera och att den skissade utvecklingen är realistisk.

Testat system

Systemet med egendeklaration har vid det här laget testats i ett par omgångar. I första skedet gjordes en utvärdering av en enda programvara. Erfarenheterna från detta försök var av värde när man senare organiserade utvärderingen av ytterligare tre programvaror. Dessa utvärderingar var mera fullödiga och bekräftar med större tyngd användbarheten hos den utvärderingsmetod som här diskuteras.

Som grund för testningen ligger de två internationella standarderna ISO 9126, som beskriver en kvalitetsmodell för mjukvaruprodukter, och ISO 14598-5, vilken behandlar själva utvärderingsmetoden.

Det var för att få mer erfarenhet av den nya metoden som SPI i januari 1999 gick ut med en förfrågan till sina medlemmar om några av dem frivilligt ville ställa programvaruprodukter till förfogande för utvärdering. I förutsättningarna ingick att de, utöver själva programvaran, utan kostnad skulle ställa upp med vissa arbetsinsatser.

En handfull företag anmälde intresse och så småningom kom SPI överens med att genomföra utvärderingen av tre programvaror tillsammans med de tre företag som tagit fram produkterna. De tre företagen och produkterna är:

ZoomON AB, Stockholm, med programvaran *ZoomON Redliner ver 1.6 Prerelease 2*.

ZoomON Redliner är en webb-baserad programvaran som gör det möjligt för auktoriserade användare uppkopplade på Internet, att titta på CAD-ritningar, kommentera och komplettera dessa på olika sätt. Föreslagna ändringar läggs in på ett överlägg och kan tas bort eller förändras. Den ursprungliga CAD-ritningen förändras inte. Själva programvaran lig-

ger lagrad på en server och är plattformsoberoende. Man kan bl a använda PC, Mac och UNIX-datorer.

Inobiz AB, Stockholm med programvaran Inobiz Development System.

Inobiz Development System är avsedd att användas vid elektronisk handel och kan utnyttjas för att skapa kopplingar mellan olika typer av system, exempelvis ERP, webbhandelsplatser och säljstödssystem. Kopplingarna kan, men behöver inte vara, baserade på EDIFACT standarden.

Benefit AB, Karlstad bidrar med programvaran ECS Check Out.

ECS Check Out är en kassamodul som ingår i ett större programvarustandardssystem för handel. Modulen hanterar elektroniskt de registreringsuppgifter som normalt görs i vanliga kassaterminaler. Modulen kan användas såväl i vanliga butikskassor som vid elektronisk handel, t ex på Internet.

De tre företagen har utan ersättning aktivt deltagit i själva utvärderingsarbetet. Utvärderingen har sedan genomförts med utgångspunkt från de internationella standarderna som nämnts tidigare.

Syftet har varit att visa att den nya metoden är tillämpbar, funktionsduglig och möjlig att använda på olika typer av programvaror. I princip kan metoden brukas på alla programvaror även om den i första hand förmodligen kommer att tillämpas på programvaror med rimlig komplexitet och omfattning. Det är emellertid inget som hindrar att man enligt samma metod utvärderar även stora och komplexa programvaror, under förutsättning att man vill avsätta de resurser som behövs för detta.

Testningen av de tre programvarorna som vi här talar om påbörjades i februari månad 1999.

För att få en uppfattning om hur den praktiska tillämpningen gått till ska vi följa utvärderingen av ett av de tre företagen och dess produkt, nämligen ZoomON och programvaran ZoomON Redliner.

Kontrakt med SPI

Innan utvärderingsarbetet startade på allvar skrevs ett kontrakt mellan SPI och ZoomON. Företaget bekräftar där att det ställer sin programvara till förfogande för utvärdering. I avtalet förbinder sig företaget att acceptera resultaten och att utvärderingen i oidentifierad form får utgöra underlag för revidering av SPI:s Manual för utvärdering av programvaror. Vidare lovade företaget att utan kostnad medverka med max 12 mandagar i utvärderingen.

SPI kommer ju i fortsättningen att spela en viktig roll som garant och administratör för den databas där alla utvärderingar kommer att redovisas. SPI är dessutom initiativtagare till metodens utveckling och tillika ansvarig för att de första testerna nu kunnat genomföras. Därför krävs det vissa juridiska avtal med de företag som utvärderat sina produkter.

I avtalet, som är skrivet på engelska, finns bl a angivet karakteristiken av utvärderingsprocessen enligt följande:

Characteristics of the evaluation process (5.3 i ISO 14598-5)

Repeatability: The evaluation is made using experienced technique in document and project control.

Reproducibility: The evaluation of the product follows ISO/IEC FDIS 9126-1, 1998-12-03 and ISO/IEC 14598-5, 1998-07-01.

Impartiality: The evaluators have no financial or marketing interest in the evaluated product.

Objectivity: The evaluation of the product follows ISO/IEC FDIS 9126-1, 1998-12-03 and ISO/IEC 14598-5, 1998-07-01 and therefore the evaluators are forced into objective assessments due to the wide range of participants in the evaluation group.

Den ovanstående texten säger i huvudsak två saker. Dels att testerna ska utföras på ett sådant sätt att de är: repeterbara, reproducerbara, opartiska och objektiva. Dels hänvisar man till de internationella standarder som utgör grunden för utvärderingen.

Projektgrupp

I samband med att avtalet utarbetades, bildades också en projektgrupp. Projektledare för denna grupp har varit Lennart Piper, som är knuten till SPI/SINF. Till sin hjälp har han haft Mats Ran, kvalitetsspecialist på varor och verksam vid Quality Laboratories Sweden AB. Dessutom har Björn Rosenquist från SEMKO följt projektet med avsikt att senare kunna starta certifieringsverksamhet. Samma personer har ingått även vid utvärdering av de övriga två programvarorna.

Genom avtalet har parternas åtaganden reglerats och därmed kunde utvärderingen påbörjas enligt de riktlinjer som står angivna i Manual för utvärdering av programvara. Under punkt 4.2 sägs bl a att "Utvärderingen

av en programvara sker när uppdragsgivaren för utvärderingen begär en sådan av utvärderaren.

Fem utvärderingsaktiviteter

Därefter har utvärderingsgruppen i princip följt de fem utvärderingsaktiviteter som gäller vid utvärdering och som något berörts i föregående kapitel. Den första aktiviteten handlar om att fastställa *utvärderingskraven*. I manualen sägs bl a att man ”analyserar uppdragsgivarens krav för utvärderingen för att identifiera de verkliga kraven för utvärderingen.” Det innebär i praktiken att man kommer överens om vilka utvärderingskrav som ska gälla för det fortsatta arbetet.

Denna punkt uppfylldes i det aktuella fallet genom att själva projektgruppen träffade representanter för produktägaren för att gå igenom programvaran. Under dessa möten försökte utvärderingsgruppen lära sig så mycket som möjligt om programvaran, dess funktion, vilka uppgifter den var avsedd att lösa etc.

Med detta som underlag plus annan tillgänglig information om programvaran blev det sedan möjligt för utvärderingsgruppen att utarbeta en ”*utvärderingsspecifikation* grundad på utvärderingskraven och beskrivningen av produkten tillhandahållen av uppdragsgivaren.”

I och med detta hade man tagit det andra steget av fem.

Utvärderingsnivå

Ett problem som man vid varje utvärdering bör fundera något över är vilken utvärderingsnivå man ska välja. Utvärderingsnivån säger en del om hur djupt man vill utföra sina mätningar och värderingar av de olika egenskaperna, men den kommer också att styra de metoder och instrument som behövs för att genomföra utvärderingen på den nivå man valt.

Väljer man en djup nivå måste man ha motsvarande instrument och metoder som med rätt noggrannhet verkligen mäter på den detaljnivå man valt. Tar man t ex en liknelse med en digital termometer som visar temperaturen i flera decimaler så är ju denna exakta presentation inte bara meningslös utan direkt vilseledande, om själva givaren, med vars hjälp själva mätningen görs, bara har en noggrannhet motsvarande exempelvis en decimal.

Nivån måste också stå i viss relation till programvarans användning. Därför tittade man till att börja med bl a på de tänkta målgrupperna. I ZoomON:s

fall har dessa definierats som ”stora företag/administratörer, stora företag/betraktare, stora företag/kommenterare, leverantörer till stora företag/administratörer, leverantörer till stora företag/betraktare, leverantörer till stora företag/kommenterare, ZoomON/utvecklare, ZoomON/support.”

Därtill måste man ta hänsyn till den miljö programvaran är tänkt att användas i. Användningen kan ha samband med trygghet, säkerhet, ekonomi och tillämpning. Om en programvara ska användas i en miljö där driftstörningar, till följd av programvarufel, kan få stora konsekvenser måste naturligtvis kraven på programvaran vara högre än om ett programvarufel ”bara” innebär att datorn måste startas om.

Utvärdering på olika nivåer kan därför sägas ge olika nivåer av tillförlitlighet vad det gäller kvaliteten hos programvaran.

I den svenska manualen för utvärdering av programvaror föreslår man fyra övergripande utvärderingsnivåer: A, B, C och D. Nivåerna är uppbyggda i hierarkier med A som den högsta nivån och D som den lägsta. På nivå A tillämpas de mest stringenta utvärderingsteknikerna som därmed också ger den högsta tillförlitligheten. Den lägsta nivån är D.

Utvärderingsnivån kan vara olika för olika kvalitetsegenskaper.

	Nivå D	Nivå C	Nivå B	Nivå A
Funktion (Funktionalitet)	Funktionell provning (black box)	Granskning (checklista)	Komponentprovning (glass box)	Formellt bevis
Tillförlitlighet	Programmeringsspråk hjälpmedel	Analys av feltoleransen	Pålitlighet tillväxtmodell	Formellt bevis
Användbarhet	Inspektion av gränssnitt	Överensstämmelse mot gränssnitts standard	Provning i laboratorium	Användarens mentala modell
Effektivitet	Mätning av exekveringstid	Benchmarktestning	Algoritmisk komplexitet	Analys av utförandeprofil
Underhållbarhet	Granskning av dokument (checklistor)	Statisk analys	Analys av utvecklings processen	Utvärdering av spårbarhet
Flyttbarhet	Analys av installation	Överensstämmelse mot programmeringsregler	Utvärdering av omgivningens begränsningar	Utvärdering av programplanering

På nivå A tillämpas de mest stringenta utvärderingsteknikerna. D är den lägsta nivån.

De olika nivåerna kan relateras till de konsekvenser som kan uppstå vid felaktigheter på de valda nivåerna. Riskvärderingen kan omfatta trygghet, ekonomi, säkerhet, miljö och marknadsföring. Nedanstående är ett exempel hur man kan ange konsekvenserna för olika nivåer

Nivå D	Liten skada på egendom, ingen risk för människor
Nivå C	Skada på egendom. Få människor skadade
Nivå B	Hot om människors liv
Nivå A	Många människor dödade

Konsekvenser på fyra olika nivåer.

När utvärderarna har bestämt sig för utvärderingsnivå och har klarat av att göra en utvärderingsspecifikation, som utgjorde det andra steget, kan man övergå till nästa moment.

Utvärderingsplan

I det tredje steget tar man fram en ”*utvärderingsplan* grundad på utvärderingsspecifikationen, komponenterna för programvaran som utvärderas och utvärderingsmetoderna föreslagna av utvärderaren.”

Med utvärderingsmetod menas en rutin ”som beskriver de åtgärder som utförs av utvärderaren för att erhålla resultaten för den specificerade mätningen eller veriferingen som tillämpas på de specificerade produktkomponenterna eller på produkten som helhet.”

För att kunna genomföra ovanstående punkt måste man arbeta fram kravnivåer för de olika egenskaper som ska undersökas. Vid utvärderingen av programvaror utgår man alltså från de sju huvudegenskaperna vi tidigare har berört. Varje huvudegenskap förtydligas sedan med hjälp av ett antal underenskaper.

Huvudegenskaper och underenskaper finns generellt beskrivna i ISO 9126-1. Som exempel citeras här den första huvudegenskapen och en av underenskaperna, såsom de beskrivs på engelska i den nyss refererade standarden inklusive några anmärkningar:

Functionality

The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.

NOTE 1 This characteristic is concerned with what the software does to fulfil needs, whereas the other characteristics are mainly concerned with when and how it fulfils needs.

NOTE 2 For the stated and implied needs in this characteristic, the note to the definition of quality in B.21 applies).

NOTE 3 For a system which is operated by a user, the combination of functionality, reliability, usability and efficiency can be measured externally by quality in use (see clause 7).

1.1 Suitability

The capability of the software product to provide an appropriate set of functions for specified tasks and user objectives.

NOTE 1 Examples of appropriateness are task oriented composition of functions from constituent sub-functions, and capacities of tables.

NOTE 2 Suitability corresponds to suitability for the task in ISO 9241-10

För varje undergenskap sätter man ett grundkrav som säger vad som ska göras eller inte göras eller vad egenskapen handlar om.

Underenskaperna förklaras med hjälp av attribut, som så entydigt som möjligt beskriver en begränsad del eller funktion av produkten och dess egenskaper. Det är således med hjälp av attributen som man beskriver en viss funktion och det är attributen som ligger till grund för de mätningar och värderingar man sedan gör av programvarans olika underenskaper.

De viktiga attributen

Attributen ska vara relaterade till grundkraven. Det är inte säkert att man omedelbart lyckas med att få en rak koppling mellan attribut och grundkrav. Ibland krävs det att man i diskussionerna går fram och tillbaka mellan grundkrav och produktattribut. Så småningom blir det möjligt att formu-

lera en kolumn med generella grundkrav motsvarande programvarans 31 underegenskaper. Sedan man diskuterat och slipat på formuleringarna kan man också formulera alla de produktattribut som krävs för att kunna gå vidare. Det var ungefär så utvärderingsgruppen gjorde.

Formuleringen av de olika attributen hör till det viktigaste i hela utvärderingsprocessen. Här gäller det att tänka klart och koncist så att man så exakt och otvetydigt som möjligt beskriver vad man vill utvärdera och mäta. Arbetet underlättas av om man i gruppen kan testa de olika formuleringarna emot varandra. Övning ger efter hand färdighet och bättre kunskaper och därmed också bättre slutresultat.

Inte bara attributens formulering är viktig utan också antalet. Ju fler attribut man använder, desto fler tester måste man göra vilket leder till bättre och tydligare egenskaper och resultat.

På sikt kan man tänka sig att arbetet förenklas med hjälp av databaser, där beskrivningen av olika attribut lagras. Databaserna är tänkta att byggas upp efterhand och kommer så småningom att utgöra ett bra underlag för formulering av attribut vid utvärdering av nya programvaror.

I den aktuella utvärderingen skapade man en tabell med de sju grundkraven i en kolumn längst till vänster. Därefter fyllde man på med en ny kolumn till höger, där underegenskaperna för varje huvudegenskap listades. I nästa kolumn skrev man in grundkravet för varje underegenskap. Därefter följde formuleringen av de olika produktattributen.

Eftersom allt man gör måste vara spårbart är det viktigt att ge varje formulering, åtgärd etc en referens eller adress, med vars hjälp man kan gå tillbaka och se var uppgiften kommer ifrån. I det aktuella fallet har detta gjorts med sifferbeteckningar enligt modellen 1.1.1.1; 1.1.1.2 osv.

Med detta som bakgrund kan vi presentera hur en del av testtabellen ser ut för ZoomON:s programvara.

SPI – Föreningen Svensk Programvaruindustri

Projekt EVAS – Evaluation of Software

Målgrupper/Roller för ZoomOn

stora företag/administratör

stora företag/betraktare

stora företag/kommenterare

leverantörer till stora företag/administratör

leverantörer till stora företag/betraktare

leverantörer till stora företag/kommenterare

ZoomOn/utvecklare

ZoomOn/support

Egendeklaration av programvaror

Egen-skaper	Underegen-skaper	Grundkrav	Veri-fikat	Produktattribut
1 Func-tionality	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.1 skriva ut en zoom bild	
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.2 zooma	
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.3 färg till svart/vit	
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.4 byta bakgrundsfärg	
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.5 rita & skriva	
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.6 färgsätta	
	1.1 Suitability	1.1.1 hantera grundfunktioner	1.1.1.7 font storlek	
	1.3 Interoperability	1.3.1 hantera basutrustning och gränssnitt	1.4.1.1 ritningsdata skall ej kunna manipuleras	
	1.4 Security	1.4.1 grundläggande datasäkerhet	1.5.1.1 uppfylla HPGL	
	1.5 Compliance	1.5.1 stöd för branschstandarder		
2 Relia-bility	2.2 Fault tolerance	2.2.1 grundläggande felhantering	2.2.1.1 meddelande vid ej understött format	
	2.2 Fault tolerance	2.2.1 grundläggande felhantering	2.2.1.2 vägledande felmeddelanden	
	2.2 Fault tolerance	2.2.2 hantera läsa och tolka data	2.2.2.1 korrekt återgivning	
	2.3 Recoverability	2.3.1 ej förlora data		
3 Usa-bility	3.1 Understandability	3.1.1 användaredokumentation	3.1.1.1 indexsökning	
	3.1 Understandability	3.1.1 användaredokumentation	3.1.1.2 innehållsförteckning	
	3.1 Understandability	3.1.2 konfigurerbar	3.1.2.1 användarespecifika konfigureringar	
	3.2 Learnability	3.2.1 användningsexempel	3.2.1.1 det skall finnas dokumenterade exempel på typfall	
	3.2 Learnability	3.2.1 användningsexempel	3.2.1.2 snabbkurs i Java och HTML	
	3.2 Learnability	3.2.2 hjälpfunktioner	3.2.2.1 on-line help redraw??	
	3.2 Learnability			
	3.3 Operability	3.3.1 grundläggande operativa funktioner	3.3.1.1 knyta ihop produkt med ritningar	
	3.3 Operability	3.3.1 grundläggande operativa funktioner	3.3.1.3 ångra	
	3.4 Attractiveness	3.4.1 lättförstålig	3.4.1.1 användaredokumentation	
3.5 Compliance	3.5.1 följer användaregränssnittstandarder	3.5.1.1 följer web/windows standard		
	3.5 Compliance	3.5.1 följer användaregränssnittstandarder	3.5.1.2 kort kommandon	
	3.5 Compliance	3.5.2 resultat följer branschstandarder		
4 Effi-ciency	4.1 Time behaviour	4.1.1 acceptabla svarstider	4.1.1.1 svarstid för användaren	
	4.2 Resource utilisation	4.2.1 maskinvaruutnyttjande	4.2.1.1 kompakt	
5 Main-tainability	5.1 Analysability	5.1.1 programvarudokumentation		
	5.2 Changeability	5.2.1 uppdatering av befintlig produktinstallation	5.2.1.1 uppdateringstid	
6 Port-ability	6.1 Adaptability	6.1.1 funktionalitet för olika plattformar	6.1.1.1 plattformsoberoende avseende operativsystem	
	6.1 Adaptability	6.1.1 funktionalitet för olika plattformar	6.1.1.2 kunna hantera olika filformat	

	6.1 Adaptability	6.1.1 funktionalitet för olika plattformar	6.1.1.3 kunna spara i olika format
	6.2 Installability	6.2.1 lätt att installera	6.2.1.1 installationsanvisningar
	6.2 Installability	6.2.1 lätt att installera	6.2.1.2 tid för driftsättning
	6.5 Compliance	6.5.1 följa språkstandard	6.5.1.1 localization
7 Quality in use	7.1 Effectiveness	7.1.1 flexibelt arbetssätt	
	7.2 Productivity	7.2.1 tid för att nå önskat resultat	
	7.4 Satisfaction	7.4.1 produktinlärning	

Egenskaper, underenskaper, grundkrav och produktattribut.

Tabellen ovan har kolumner för egenskaper, huvudegenskaper, grundkrav och produktattribut. För en del grundkrav krävs det mer än ett attribut och man kan därför bygga på med ännu en kolumn till höger för flera attribut.

Till höger om kolumnen eller kolumnerna för produktattribut tillkommer ytterligare en kolumn, nämligen kravnivåer för varje produktattribut.

Kravnivåer

Även kravnivåerna till de olika attributen kan vara svåra att definiera. Den nivå man bestämmer sig för har bl a att göra med vilka krav och förväntningar man har på produkten. Hur mycket bör eller ska produkten klara av? Vad förväntar sig kunden att programvaran ska kunna utföra och hantera? Hur ser det ut på marknaden i övrigt för den här typen av programvaror?

Sätter man för låg nivå kanske man visserligen får full pott i utvärderingen, dvs resultatet ser mycket bra ut. Men den som studerar utvärderingen kan ju också se vilken låg nivå som valts och då är ett "bra" resultat inte värt så värst mycket som det vid första ögonkastet kan tyckas. Tvärtom, en alltför låg kravnivå tyder på att programvaran inte klarar av speciellt mycket, att programvarutillverkaren försöker dölja något eller i varje fall inte har mycket att komma med.

Om man å andra sidan sätter en alltför hög nivå, risker man att få många deltester underkända vilket kanske kan ge indikation på att produkten ännu inte är färdig för marknaden.

Det säger sig självt att om man sätter lite högre nivå än marknaden i övrigt och sedan kan visa att programvaran också klarar av denna högre nivå, så kommer detta att utgöra en fördel när produkten marknadsförs. På så leder den här utvärderingsmetoden också till att programprodukterna utvecklas mot allt bättre kvalitet och funktion, vilket är ett av syftena.

Efter hand som vi får många utvärderade produkter inom olika produktområden kan det kanske också bli aktuellt med att sätta upp gemensamma kravnivåer för likartade produkter.

Med utgångspunkt från den del av ZoomON:s testtabell vi just har visat kan vi komplettera den med en kolumn med rubriken ”Kravnivå för produktattribut”. Till vänster om denna finns en kolumn med de tidigare produktattributen och längst till vänster visas de nummer som ger oss möjlighet att spåra informationen. I vårt fall kan dessa beteckningar också användas för att matcha nedanstående tabell med den tidigare tabellen, som innehåller underegenskaper och huvudegenskaper. Av utrymmesskäl tas bara en del av tabellen med.

ZoomON

Verifikat	Produktattribut	Kravnivå för produktattribut
1.1.1.1	skriva ut en zoom bild	Bild ska kunna skrivas ut skalenligt i alla definierade zoomlägen
1.1.1.2	zooma	Bild ska kunna zoomas i intervallet 25%-400%
1.1.1.3	färg till svart/vit	Bild ska kunna konverteras från färg till gråskala (med samma upplösning)
1.1.1.4	byta bakgrundsfärg	Bakgrundsfärg ska kunna bytas till valfri färg (som originalbild)
1.1.1.5	rita & skriva	Lista med objekt som ska kunna ritas och text som ska kunna skrivas
1.1.1.6	färgsätta	Ritade objekt ska kunna färgsättas i valfri färg (som originalbild)
1.1.1.7	fontstorlek	Fontstorlek ska kunna ändras mellan 8 till 72 punkter
1.1.1.8	hjälp	Skall finnas
1.1.1.9	Öppna, Spara, Spara som	Skall kunna göras
1.1.1.10	Importera/Exportera	Skall kunna göras för understödda format
1.1.1.11	Panorera	Skall kunna göras över 100% av ritningsytan i alla skallägen
1.1.1.12	Unquiry	Skall kunna göras
1.1.1.13	Set Unit	Skall kunna göras
1.2.1.1	skriva enligt skala	Programmet skall kunna läsa in en fil och sedan skriva ut den skalenligt
1.3.1.1	fungera i normal nätverksmiljö	Stödja alla grundfunktioner
1.4.1.1	ritningsdata skall ej kunna manipuleras	Ingen manipulering av originaldata ska kunna ske med produkten
1.5.1.1	uppfylla HPGL	Stödja alla grundkommandon i HPGL
2.2.1.1	meddelande vid ej understött format	Alla ej understödda format ska ha ett informationsmeddelande
2.2.1.2	vägledande felmeddelanden	Alla felmeddelanden ska innehålla en vägledande text om vad som är fel och dess påverkan
2.2.2.1	korrekt återgivning	Alla ritningsdata ska återges korrekt
2.3.1.2	ej förlora sparade data	Data i en sparad fil ska ej gå förlorad vid avbrott
3.1.1.1	indexsökning	Användardokumentation ska ha funktion för indexsökning och denna ska täcka 80% av alla funktioner
3.1.1.2	innehållsförteckning	Användardokumentation ska ha innehållsförteckning

ZoomOn/support

Verifikat	Produktattribut	Kravnivå för produktattribut
1.1.1.1	skriva ut en zoom bild	Bild ska kunna skrivas ut skalenligt i alla definierade zoomlägen
1.1.1.2	zooma	Bild ska kunna zoomas i intervallet 25%-400%
1.1.1.3	färg till svart/vit	Bild ska kunna konverteras från färg till gråskala (med samma upplösning)
1.1.1.4	byta bakgrundsfärg	Bakgrundsfärg ska kunna bytas till valfri färg (som originalbild)
1.1.1.5	rita & skriva	Lista med objekt som ska kunna ritas och text som ska kunna skrivas
1.1.1.6	färgsätta	Ritade objekt ska kunna färgsättas i valfri färg (som originalbild)
1.1.1.7	fontstorlek	Fontstorlek ska kunna ändras mellan 8 till 72 punkter
1.1.1.8	hjälp	Skall finnas
1.1.1.9	Öppna, Spara, Spara som	Skall kunna göras
1.1.1.10	Importera/Exportera	Skall kunna göras för understödda format
1.1.1.11	Panorera	Skall kunna göras över 100% av ritningsytan i alla skallägen
1.1.1.12	Unquiry	Skall kunna göras
1.1.1.13	Set Unit	Skall kunna göras
1.2.1.1	skriva enligt skala	Programmet skall kunna läsa in en fil och sedan skriva ut den skalenligt
1.3.1.1	fungera i normal nätverksmiljö	Stödja alla grundfunktioner
1.4.1.1	ritningsdata skall ej kunna manipuleras	Ingen manipulering av originaldata ska kunna ske med produkten
1.5.1.1	uppfylla HPGL	Stödja alla grundkommandon i HPGL
2.2.1.1	meddelande vid ej understött format	Alla ej understödda format ska ha ett informationsmeddelande
2.2.1.2	vägledande felmeddelanden	Alla felmeddelanden ska innehålla en vägledande text om vad som är fel och dess påverkan
2.2.2.1	korrekt återgivning	Alla ritningsdata ska återges korrekt
2.3.1.2	ej förlora sparade data	Data i en sparad fil ska ej gå förlorad vid avbrott
3.1.1.1	indexsökning	Användardokumentation ska ha funktion för indexsökning och denna ska täcka 80% av alla funktioner
3.1.1.2	innehållsförteckning	Användardokumentation ska ha innehållsförteckning
3.1.2.1	användarspecifika konfigurationer	100% av de grundläggande funktionerna ska kunna konfigureras per användare
3.1.2.1	användarspecifika konfigurationer	100% av de grundläggande funktionerna ska kunna konfigureras per användare

Produktattributen tillsammans med motsvarande kravnivåer:

Huvudegenskapen i tabellen ovan är Funktion eller Functionality och är märkt 1. Huvudegenskapens första underregenskap är Lämplighet eller Suitability märkt 1.1. Till denna underregenskap hör ett stort antal grundkrav som alla kan betecknas med "hantera grundfunktion" och får beteckningen 1.1.1.

Så långt har vi följt grundförutsättningarna som gäller för varje programvara. Nu återstår att man för varje grundfunktion formulerar vad programvaran ska utföra. Detta görs med ett attribut, där man med några

rader beskriver vad som utförs. Vi tar som exempel attributet ”skriva ut en zoombild”, som får beteckningen 1.1.1.1.

Mot detta attribut sätter vi en kravnivå som så exakt som möjligt beskriver de krav som bör uppfyllas för att utskrift av en zoombild kan anses godtagbar. Kravnivån kan ses som ett förtydligande av attributet och samtidigt som en miniminivå för vad programvaran ska kunna utföra. I det aktuella fallet beskrivs kravnivån till attributet 1.1.1.1 på följande sätt: ”Bild ska kunna skrivas ut skalenligt i alla definierade zoomlägen”.

Går vi vidare till attributet 1.1.1.2 ”zooma” så är kravnivån beskriven på följande sätt: ”Bild ska kunna zoomas i intervallet 25-40%.”

För det följande attributet, 1.1.1.3 ”färg till svart/vit” är kravnivån: ”Bild ska kunna konverteras från färg till gråskala (med samma upplösning)”.

På likande sätt går man igenom alla övriga attribut tills man har en komplett lista som omfattar alla sju huvudegenskaperna och de 31 under-egenskaperna.

I förutsättningarna sägs att varje egendeklaration ska omfatta minst 22 definierade och dokumenterade produktattribut. För varje produktattribut ska man definiera och dokumentera:

- grundkrav
- kravnivå för varje enskilt produktattribut
- verifikat
- verifieringsmetod

Metoder

För att kunna utföra testerna behövs också ett antal väl definierade metoder. De metoder som användes i det här sammanhanget har tagits fram av utvärderingsgruppen och bygger på erfarenheter från det internationella standardiseringsarbetet. I samband med att mät- och testresultaten offentliggörs beskrivs också de mätmetoder som används.

Framtagning av mät- och utvärderingsmetoder pågår även på internationell nivå. Detta arbete kommer att ta flera år i anspråk innan man har täckning för alla situationer.

Ett exempel på hur en metod kan beskrivas visas nedan. Här har man valt den lägsta nivån, dvs nivå D och metoden är i detta fall avsedd att testa huvudegenskapen ”användbarhet” (usability) samt under-egenskapen ”försåtlighet” (understandability). Till metoden hör en förklaring hur mätresultaten ska tolkas.

Metod O. Inspektion av dokumentation

Inledning

Denna utvärderingsmodul skall användas vid bedömning av om en programvaras tillgängliga funktioner kan konfigureras per användare.

Omfattning

Egenskap	Usability, underregenskap understandability
Nivå	D
Teknik	Test av funktionallitet
Användningsområde	Generellt vid kontroll av konfigureringsmöjligheter hos en programvara

Referenser

Definitioner

Definition	Beskrivning
Användarspecifik konfigurerings	Förmåga hos en programvara att anpassa tillgänglig funktion beroende på vem som är aktuella användaren utan inblandning från användaren

Indata till utvärderingen

Data	Beskrivning
Indata	Följande källor används som indata till utvärderingen: Aktuell dokumentation Aktuell programvara och version

Tolkning av resultat

Tolkning av mätetal	Godkänt om en programvaras funktion kan konfigureras enligt ställda krav.
Rapportering	Följande information skall rapporteras som resultat av utvärderingen: Identifiering av använd indata. Lista med funktioner som ska vara konfigurerbara. Nivå för uppfyllning av krav.

Ytterligare en metod som används visas nedan. Här handlar det om inspektion av systemlösning och i detta fall har vi flyttat upp till nivå C.

Metod G. Inspektion av systemlösning

Dokumentationen av denna metod följer ISO/IEC CD 14598-6.

Förord

Ej tillämpligt

Inledning

Den här utvärderingsmodulen ska användas vid inspektion av en programvaras systemlösning.

Omfattning

Egenskaper	Funktionalitet, Tillförlitlighet, Underhållsmässighet och Flyttbarhet
Nivå	Nivå C
Teknik	Genom att inspektera konstruktionsdokumentation och intervjua/diskutera specifika systemlösningar med insatt utvecklingspersonal
Användnings område	Generellt vid inspektion av en programvaras systemlösning med syfte att uppfylla en specifik egenskap

Referenser

Inga referenser

Definitioner

Följande definitioner gäller i denna utvärderingsmodul:

Term	Definition
Egenskap	En programvaras förmåga att hantera en specifik uppgift/funktionalitet

Indata till utvärderingen

Data	Beskrivning
Indata	Följande källor används som indata till utvärderingen <ul style="list-style-type: none">• Produktkomponent: Användardokumentation• Utvecklingspersonal• Programvara• Krav = Egenskap hos en specifik systemlösning

Data	Följande måste mätas för att kunna utföra utvärderingen enligt föreslagna metodik <ul style="list-style-type: none">• A = Prestanda hos en egenskap
Mätetal och deras relation	–
Tolkning av resultat	
Bearbetning av mätetal	–
Tolkning av mätetal	<ul style="list-style-type: none">• Godkänt om A uppfyller Krav• Underkänt om A ej uppfyller
Rapportering	Följande information ska rapporteras som resultat av utvärderingen: <ol style="list-style-type: none">1. Identifiering av använda användardokument2. Identifiering av intervjuad utvecklingspersonal3. Identifiering av använda programvaror4. Prestanda hos systemlösning5. Motsvarande betyg

Med de båda mätmetoderna ovan som exempel är vi nu framme vid det fjärde steget som innebär att man *genomför utvärderingsplanen*. I detta ingår bl a inspektion av programvaran vilken vi redan ha klarat av. Även modellering skulle kunna ingå i detta moment, men i vårt fall handlar det om mätning och testning av produkten enligt planen. Under testningen ska resultaten av alla aktiviteterna och mätningarna redovisas och skrivs ner i ett preliminärt protokoll.

Rent allmänt gäller att den praktiska utvärderingen bör göras där man kan arbeta ostört och där det finns tillräckliga resurser i form av datorer och annan utrustning som är nödvändig för att mätningarna ska kunna utföras tillfredsställande. I ZoomON:s fall skulle t ex den testade programvaran kunna användas på olika plattformar (PC, Mac, UNIX). Då bör man också kunna göra testerna på de olika plattformarna.

Två dagars testning

Testningarna och mätningarna var planerade att ta två dagar i anspråk för var och en av de tre programvarorna och genomfördes i juni och juli 1999.

I ZoomON:s fall gjordes testerna dels ute på företaget, dels i SPI:s lokaler. Den metodik som följdes finns beskriven i ISO 14590-5.

Som testprotokoll använde testgruppen i stor utsträckning Excel-blad, där uppgifterna efter hand fördes in direkt via datorn. Man kan också göra fortlöpande anteckningar på papper om det är lämpligare. Oavsett hur man gör är det viktigt att man omedelbart efter mätningarna noterar resultaten och eventuella anmärkningar samt håller ordning på vad man gör. Systematik är en självklarhet och man måste hela tiden ha i minnet att allting som görs ska vara spårbart så att de olika momenten i princip kan återupprepas.

Utvärdering i grupp

Det är en klar fördel, ja, kanske rent utav en förutsättning för ett bra resultat, om flera personer deltar i testningen. En stor del av arbetet handlar om att göra bedömningar. Består testgruppen av flera personer med lite olika erfarenheter får man en bättre belysning av mätningarna än om man skulle agera ensam.

Tre saker avgör mätningens kvalitet:

- kunskaperna hos de personer som gör bedömningen. (En testgrupp bör bestå av minst tre personer);
- kriterier på vad som är styrande;
- metodik, som används; i detta fall följer man ISO 14590-5

Lämpligen bör det i testpanelen ingå någon eller några personer med erfarenhet av just utvärderingar. I det aktuella fallet var erfarenheterna av just den här mätmetoden mycket begränsad eftersom den är alldeles ny i sin tillämpning. Däremot kunde man utnyttja personer med erfarenheter av andra, liknande mätmetoder och utvärderingar, vilket borgade för väl genomfört arbete och slutresultat.

På sikt är tanken att man genom utbildning ska kunna skapa bra förutsättningar för att så många som möjligt ska få möjlighet att utvärdera programvaror enligt de riktlinjer som utvärderingsmetoden anvisar.

Utvärderingen av de tre aktuella programvarorna har gjorts under ledning av Lennart Piper och Mats Ran. Den förstnämnde har arbetat mycket med både mätningar och standardiseringsarbete, den sistnämnde ha lång erfarenhet av mätningar och kvalitetsarbete. Utöver dessa personer har dessutom respektive företag ställt egen personal till förfogande, där deras kunskaper om programvaran naturligtvis också är viktigt.

Fortlöpande redovisning

Under två dagar sitter alltså tre personer och utför mätningar och värderingar av ZoomON:s programvara ZoomON Redliner. Vad som utfördes framgår av de produktattribut som finns formulerade för varje underegen-skap och vad som är godkänt framgår av den kravnivå som är definierad och redovisad till höger om produktattributen. Till detta har också olika mätmetoder tagits fram.

Resultaten redovisas fortlöpande under mätningens gång och kopplas till en unik sifferbeteckning för varje mätning. När resultaten väl är utlag-da i den tilltänkta databasen eller motsvarande kan man via Internet klicka på den understrukna sifferbeteckningen. Bakom denna döljer sig verifika-tet, som för varje mätning ger upplysningar om vad som har gjorts och vilken metod som har använts.

Låt oss klicka på några av dessa verifikat och se vad mätningen har gett. Vi väljer att klicka på verifikat 1.1.1.1 till och med 1.1.1.13. De berör alla underegen-skapen Lämplighet (Suitability), som i sin tur hör till huvud-egen-skapen Funktion (Functionality).

Utvärderingsverifikat

ZoomON
ZoomON Redliner
EVAS

Förutsättningar

Utvärderare: Ola Andersson
Datum: 991108

Utvärderat attribut

- 1.1.1.1 Hantera grundfunktioner, skriva ut
- 1.1.1.2 Hantera grundfunktioner, zooma
- 1.1.1.3 Hantera grundfunktioner, färg till svartvitt
- 1.1.1.4 Hantera grundfunktioner, byta bakgrundsfärg
- 1.1.1.5 Hantera grundfunktioner, rita & skriva
- 1.1.1.6 Hantera grundfunktioner, färgsätta
- 1.1.1.7 Hantera grundfunktioner, fontstorlek
- 1.1.1.8 Hantera grundfunktioner, hjälp

- 1.1.1.9 Hantera grundfunktioner, Öppna, Spara, Spara som
- 1.1.1.10 Hantera grundfunktioner, Importera/Exportera
- 1.1.1.11 Hantera grundfunktioner, Panorera
- 1.1.1.12 Hantera grundfunktioner,
- 1.1.1.13 Hantera grundfunktioner,

Del av utvärderingsverifikat som visar ett antal attribut.

Själva verifikaten inledes med namnen på programvaran respektive den som gjort utvärderingen samt datum när utvärderingen ägde rum. Sedan räknas de olika attributen upp inklusive sifferbeteckningarna för var och en, totalt 13 stycken. Till samtliga dessa har man använt mätmetod T.

Mätmetod T är beskriven på följande sätt.

Metod T. Grundfunktioner

Inledning

Alla grundfunktioner skall fungera på önskat operativsystem och version för den testade programvaran.

Omfattning

Egenskaper	Functionality – Suitability
Nivå	Nivå D
Teknik	Praktisk utföra grundfunktionerna på vald plattformar
Användningsområde	Vid normal användning

Referenser

–

Definitioner

Följande definitioner gäller i denna utvärderingsmodul:

Definition	Beskrivning
Indata till utvärderingen	
Data	Beskrivning
Indata	<ul style="list-style-type: none">• Aktuell version av programvara• Dator med måloperativ installerat
Data	Funktion ska fungera enligt definierade nivåer
Mätetal och deras relation	–

Tolkning av resultat

Bearbetning av mätetal	–
Tolkning av mätetal	Godkänt om funktion fungerar enligt ställda krav.
Rapportering	Följande information skall rapporteras som resultat av utvärderingen: Identifiering av typ av dokumentation. Metod för uppfyllande av krav.

Mot de nyss uppräknade attributen ställs de olika kravnivåer man fastställt. I nedanstående uppräkning används endast den sista siffran som referens. Efter siffrorna kommer det som kanske är intressantast i sammanhanget, nämligen redovisningen av mätresultaten i form av rådata. Några av dessa resultat kommenteras efter själva uppräknigen nedan.

Indata till utvärderingen

Tillgängliga produktkomponenter RedLiner, version 1.60

Krav (attributnivå)

- 1) Hantera grundfunktioner, skriva ut
 - a) Bild ska kunna skrivas ut skal enligt i alla definierade skallägen
- 2) Hantera grundfunktioner, zooma
 - a) Bild ska kunna zoomas i skallägen 25%-400%
- 3) Hantera grundfunktioner, färg till svartvitt
 - a) Bild ska kunna konverteras från färg till gråskala (med samma upplösning)
- 4) Hantera grundfunktioner, byta bakgrundsfärg
 - a) Bakgrundsfärg ska kunna bytas till valfri färg (som originalbild)
- 5) Hantera grundfunktioner, rita & skriva
 - a) Följande figurer och text ska kunna ritas
 - i) Box, Circle, Cloud
 - ii) Text (alla skrivtecken + siffror) på valfri position inom ritning
- 6) Hantera grundfunktioner, färgsätta
 - a) Ritade objekt ska kunna färgsättas i valfri färg (som originalbild)
- 7) Hantera grundfunktioner, fontstorlek
 - a) Fontstorlek ska kunna ändras mellan 5-72 punkter
- 8) Hantera grundfunktioner, hjälp
 - a) Ska finnas

- 9) Hantera grundfunktioner, Öppna, Spara, Spara som
 - a) Ska kunna göras
- 10) Hantera grundfunktioner, Importera/Exportera
 - a) Ska kunna göras för understödda format
- 11) Hantera grundfunktioner, Panorera
 - a) Ska kunna göras över 100% av ritningsytan i alla skallägen
- 12) Hantera grundfunktioner, Unquiry
 - a) Ska kunna göras
- 13) Hantera grundfunktioner, Set Unit
 - a) Ska kunna göras

Data som ska mätas: A1-13 = Uppfylld av kravnivå

Mätetal som ska beräknas

Rådata

I den första uppgiften ovan med beteckningen 1a) gällde det att skriva ut bild skalenligt i alla definierade skallägen. Det var uppenbarligen inga problem. Här blev resultatet otvetydigt OK.

Nästa uppgift 2a), handlar om att bilden ska kunna zoomas i skallägen 25% – 400%. Här var resultatet inte riktigt lika enkelt att tolka. I varje fall krävs det en liten anmärkning. Jodå, zoomningen går att utföra steglöst inom hela det föreskrivna området, men testaren upptäcker att det inte klart framgår inom vilket zoomområde man befinner sig. Det är visserligen inte sagt, varken i produktattributet eller i kravnivån, att man hela tiden ska se vilken zoomfaktor som gäller. Men i provningsögonblicket märker testaren att det trots allt skulle varit bra att veta, bl a för att lättare kunna kontrollera att man just uppfyller de gränser som är satta. Han gör därför en anmärkning om denna brist men slutar med att säga OK. Kravnivån är trots allt uppfylld.

Även nästa mätning, A3, kan vara värd att kommentera. Uppgiften var här att man skulle kunna ändra från färg till svartvitt. Men den funktionen ingår inte i produkten, påpekar testaren men skriver ändå OK.

Ytterligare en mätning, A4, kan ge oss intressanta synpunkter på de problem som kan uppkomma. Här ska man kontrollera om programvaran kan hantera grundfunktioner att konvertera en bild från färg till gråskala (med samma upplösning) och från färg till svartvitt. Mätningen visar att det går med viss begränsning. Begränsningen består i att det endast går att

byta till vitt, grått eller svart. Har man därmed uppfyllt kravnivån? Testaren är inte säker. Han tror det och skriver därför OK, men han lägger till ett frågetecken, sannolikt med syfte att senare diskutera resultatet med testgruppen.

Så fortsätter testaren att gå igenom även övriga uppgifter. Se nedan! Till slut kan han göra en summering. Som läsaren ser står det först att alla 13 testade attributen är godkända. På den följande raden står det emellertid att det är 12 godkända och en underkänd. Man kan anta att testaren har haft överläggningar med övriga personer som ingått i utvärderingsgruppen. Det är efter dessa överläggningar som man kommit fram till att 12 deltester är godkända, dvs de uppfyller fastställda kravnivåerna. Ett deltest är däremot underkänt.

Resultat av utvärderingen

Använda produktkomponenter
ZoomOn RedLiner, version 1.60

Rådata

- A1 = OK
- A2 = Zoom går att göra i steg samt steglöst dock syns ej aktuell zoomfaktor -> OK
- A3 = Ingår ej i produkten. Utgår -> OK
- A4 = Går endast att byta till Vit, Grå eller Svart -> OK?
- A5 = Box, Circle, Cloud ->OK, Text OK
- A6 = Kan sättas till 10 olika färger -> OK
- A7 = Kan ändras till valfri storlek i cm -> OK
- A8 = Hjälp finns via menyalternativ -> OK
- A9 = Öppna, Spara som finns, Spara finns -> OK
- A10 = Importera testad genom att 4 slumpmassigt valda filer från internet importerades -> OK
- A11 = Går att panorera i olika skallägen -> OK
- A12 = OK
- A13 = OK

Betyg: 13 Godkända

Betyg: 12 Godkända, 1 Underkänd

På liknande sätt som ovan går man igenom samtliga sju huvudegenskaper med de 31 underegenskaperna.

Till slut kan man återgå till den tidigare rapporten och i kolumnen Verifikat föra in "G" för godkänd, "U" och övriga förkortningar. Se förklaring överste på bilden. För ZoomON:s del kom resultatet att se ut på följande sätt.

Utvärderingsresultat för ZoomON Z2

SPI – Föreningen Svensk Programvaruindustri

1999-11-10/lp

Projekt EVAS – Evaluation of Software

G = Godkänd test

U = Underkänd test

E = Ej genomförd test

EG = Godkänd tilläggstest

UG = Tidigare underkänd, nu godkänd test

Verifikat	Produktattribut	Kravnivå för produktattribut
1.1.1.1 G	skriva ut en zoom bild	Bild ska kunna skrivas ut skal enligt i alla definierade zoomlägen
1.1.1.2 G	zooma	Bild ska kunna zoomas i intervallet 25%-400%
1.1.1.3 G	färg till svart/vit	Bild ska kunna konverteras från färg till gråskala (med samma upplösning)
1.1.1.4 G	byta bakgrundsfärg	Bakgrundsfärg ska kunna bytas till valfri färg (som originalbild)
1.1.1.5 G	rita & skriva	Lista med objekt som ska kunna ritas och text som ska kunna skrivas
1.1.1.6 G	färgsätta	Ritade objekt ska kunna färgsättas i valfri färg (som originalbild)
1.1.1.7 G	fontstorlek	Fontstorlek ska kunna ändras mellan 8 till 72 punkter
1.1.1.8 G	hjälp	Skall finnas
1.1.1.9 UG	Öppna, Spara, Spara som	Skall kunna göras
1.1.1.10 G	Importer/Exportera	Skall kunna göras för understödda format
1.1.1.11 G	Panorera	Skall kunna göras över 100% av ritningsytan i alla skallägen
1.1.1.12 G	Unquiry	Skall kunna göras
1.1.1.13 G	Set Unit	Skall kunna göras
1.2.1.1 G	skriva enligt skala	Programmet skall kunna läsa in en fil och sedan skriva ut den skal enligt
1.3.1.1 G	fungera i normal nätverksmiljö	Stödja alla grundfunktioner
1.4.1.1 G	ritningsdata skall ej kunna manipuleras	Ingen manipulering av originaldata ska kunna ske med produkten
1.5.1.1 UG	uppfylla HPGL	Stödja alla grundkommandon i HPGL
2.2.1.1 UG	meddelande vid ej understött format	Alla ej understödda format ska ha ett informationsmeddelande

Egendeklaration av programvaror

2.2.1.2 G	vägledande felmeddelanden	Alla felmeddelanden ska innehålla en vägledande text om vad som är fel och dess påverkan
2.2.2.1 G	korrekt återgivning	Alla ritningsdata ska återges korrekt
2.3.1.2 G	ej förlora sparade data	Data i en sparad fil ska ej gå förlorad vid avbrott
3.1.1.1 U	indexsökning	Användardokumentation ska ha funktion för indexsökning och denna ska täcka 80% av alla funktioner
3.1.1.2 G	innehållsförteckning	Användardokumentation ska ha innehållsförteckning
3.1.2.1 G	användarspecifika konfigurationer	100% av de grundläggande funktionerna ska kunna konfigureras per användare
3.2.1.1 EG	det skall finnas dokumenterade exempel på typfall	80% av grundfunktioner ska finnas som dokumenterade typfall
3.2.1.2 EG	snabbkurs i Java och HTML	Ska finnas
3.2.2.1 EG	on-line help	On-line hjälp ska finnas för 100% av alla grundfunktioner
3.3.1.1 G	knyta ihop produkt med ritningar	spårbarhet mellan produkt och ritning
3.3.1.3 G	ångra	Alla operationer ska kunna ånras (gäller ej transaktioner)
3.4.1.1 UG	användardokumentation	Användardokumentation ska täcka 100% av alla tillgängliga grundfunktioner
3.5.1.1 U	följer web/windows standard	Användargränssnitt ska följa etablerade branschstandarder (definiera)
3.5.1.2 U	kort kommandon	Ska stödja de vanligaste kortkommandona i MS Windows
3.5.2.1 G	använder etablerade branschstandarder på filformat	Utdata ska vara formaterat i ett standardsformat
4.1.1.1 G	svarstid för användaren	Svarstider ska vara acceptabla för en normal användare vid normal användning
4.2.1.1 EG	kompakt	normala maskinvarukrav för operativsystemet ska räcka för användande
5.2.1.1 G	uppdateringstid	Uppdatering skall ej innehålla manuella åtgärder
6.1.1.1 EG	plattformsoberoende avseende operativsystem	Programvara ska kunna användas på alla målplattformar utan modifiering
6.1.1.2 UG	kunna hantera olika filformat	Alla etablerade format på marknaden skall kunna hanteras
6.2.1.1 UG	installationsanvisningar	Installationsanvisning som täcker alla åtgärder som ska göras ska finnas
6.2.1.2 G	tid för driftsättning	Systemet ska kunna driftsättas inom 1 arbetsdag
6.5.1.1 EG	localization	Byte av språkrelaterade inställningar ska kunna göras och enligt standard
7.1.1.1 E	allt skall kunna göras via tangentbordet	Alla grundfunktioner ska vara åtkomliga via tangentbord
7.2.1.1 E	tid för "re-line" en ritning	Kortare än om motsvarande åtgärd skulle göras manuellt med motsvarande resultat
7.4.1.1 G	användningen av produkten skall vara intuitiv	En dator användare ska inom 10 minuter kunna använda valfri grundfunktion 90%

Minst 10+12 godkända attribut

I SPI:s bestämmelser för egendeklarationer står det i § 5:

”Varje egendeklaration skall definiera och dokumentera minst 22 produktattribut. Utgångspunkten för denna definition är de 7 egenskaperna och de 31 underegenskaperna i ISO 9126-1. För varje produktattribut skall det också definieras och dokumenteras:

- grundkrav
- kravnivå för varje enskilt produktattribut
- verifikat
- verifieringsmetod

Alla för produkten utlovade funktionaliteter enligt egenskapen funktionalitet i ISO 9126-1 med de 5 underegenskaperna skall vara verifierade och godkända. Antalet verifierade och godkända produktattribut får ej understiga 10 för egenskapen funktionalitet. För de övriga 6 egenskaperna enligt ISO 9126-1 skall minst ett produktattribut för var och en vara verifierade och godkända. Antalet verifierade och godkända produktattribut får ej understiga 12 för de övriga 6 egenskaperna enligt ISO 9126-1.”

90% attribut godkända

I det aktuella fallet med ZoomON så testade man sammanlagt 42 attribut. Av dessa godkändes 90%. Vissa tilläggstest gjordes. Detta är möjligt att göra enligt förutsättningarna men det ska i så fall också framgå i testprotokollet.

Det övergripande resultatet blev att programvaran blev godkänd. Till detta ska läggas en hel del erfarenheter. Några av dessa redovisas i följande kapitel och omfattar erfarenheter från tre företag som låtit utvärdera sina programvaror.

Eftersom programvaran uppfyllde kraven för att bli godkänd, kan företaget ZoomON använda SPI2000-märket i sin marknadsföring av just den



*SPI-märket
SPI2000.*

programvaran. Märket kan användas i samband med produktreklam och på produkter samt förpackningar till produkter. Märket ger upplysningar om att produkten har en egendeklaration och att mer information om och beskrivning av innehållet i deklarationen finns att tillgå på annat håll, t ex på Internet.

Principer för SPI 2000 och egendeklarationer

Avslutningsvis kan det vara lämpligt att citera de nio principer som alla ska följa vid användning av ISO-standarder för produktmärken och egendeklarationer.

Princip 1 – Saklig och faktabaserad information

Produktmärket SPI 2000 och egendeklarationen skall vara korrekt och verifierbar och uppfattas som relevant och icke-vilseledande.

Princip 2 – Undvikande av internationella handelshinder

Administrativa former och krav för produktmärket SPI 2000 och egendeklarationer skall inte utarbetas, antas eller användas med avsikt att skapa, eller leda till onödiga eller förtäckta internationella handelshinder.

Princip 3 – Användning av standard metoder

Produktmärket SPI 2000 och egendeklarationer skall vara baserade på tillräckligt noggranna och omfattande, om möjligt, standardiserade metoder för att garantera underlagsberäkningar och resultat som är faktabaserade och reproducerbara.

Princip 4 – Lättillgänglig information

Information om förfaringssätt för att ta fram underlagsmaterial, metoder som används och kriterier som utarbetats som underlag för produktmärket SPI 2000 och egendeklarationer skall finnas tillgänglig och på begäran till intressenter.

Princip 5 – Livscykelperspektiv

Produktmärket SPI 2000 och egendeklarationen skall beakta alla relevanta aspekter i en produkts livscykel.

Princip 6 – Stimulans av nyskapande produktutveckling

Produktmärket SPI 2000 och egendeklarationer skall inte hindra företags-specifika innovationer och andra modifieringar av produkter som har stor potential i förbättring av produktprestanda.

Princip 7 – Kostnadseffektivitet

Administrativa former för produktmärket SPI 2000 och egendeklarationer skall begränsas till ett minimum för att inte leda till höga kostnader att

ansluta sig till systemen som kan begränsa tillämpningen i små och medelstora företag.

Princip 8 – Aktiv intressentmedverkan

Arbetet att utveckla system för produktmärket SPI 2000 och egendeklarationer bör innefatta en öppen samverkan med intressenter och berörda parter i vilken ansträngningar bör göras för att uppnå konsensus.

Princip 9 – Öppenhet kring val av produktaspekter

Information om de relevanta produktaspekter som valts som underlag för kriterier för ett produktmärket SPI 2000 eller som underlag för en egendeklaration skall vara tillgängliga för inköpare och konsumenter.

4. Tre företags erfarenheter av utvärderingen

Tre företag har varit med i utvärderingen. De är därmed pionjärer på denna typ av egendeklarationer. Inget företag, varken i Sverige eller utomlands, har tidigare deltagit i något liknande, om man bortser från ett annat svenskt företag, vars programvara varit föremål för en mindre omfattande utvärdering. Erfarenheterna från denna preutvärdering låg till grund för de mera omfattande egendeklarationerna.

Vad säger då representanterna för de tre företag som varit med om utvärderingen? Vilka är deras erfarenheter? Hur har utvärderingen påverkat enskilda personer och företagen i deras syn på programkvalitet? Och vad tror de om framtiden för egendeklaration och certifiering? Här följer några synpunkter.

Varför nappade just de här tre företagen på möjligheten att vara med i utvärderingen?

De tre företag som deltog i utvärderingen beslöt sig för att vara med i försöken med egendeklaration därför att det tyckte det lät intressant, för att de gärna ville ha en genomgång av den egna programvaran, för att få synpunkter från andra än sina egna och helt enkelt för att få en opartisk genomlysning av sina respektive programvaror.

Niklas Jardeby på Inobiz uttryckte saken på följande sätt:

– Vi såg framför allt två fördelar med utvärderingen. För det första tyckte vi det skulle vara nyttigt att personer utanför det egna företaget skulle kunna ge tekniska synpunkter på vår programvara, särskilt vad det gällde gränssnitt och liknande saker. För det andra ansåg vi att en certifiering skulle vara värdefull ur marknadsföringssynpunkt.

Representanter för de andra företagen hade liknande synpunkter. Genom utvärderingen skulle man få fram svagheter i den egna programvaran så att dessa sedan skulle kunna åtgärdas före marknadsintroduktionen. Likaså framkom att det skulle vara nyttigt att lära sig något mera om standard.

Sådan kunskap kan användas för att göra en del programvarulösningarna effektivare. Genom att ta vara på vissa standardlösningar slipper man återuppfinna hjulet gång på gång.

Vilka var erfarenheterna från utvärderingsdagarna? Vad hade det för effekt på det fortsatta arbetet med företagens programvaror?

– För oss innebar de två dagarnas utvärdering att vi forcerade fram användarmanual och installationsinstruktioner, påpekade Henrik Danielsson på Benefit. Och Ola Andersson på ZoomON kompletterar på följande sätt:

– Vi har blivit mer medvetna om vad man bör tänka på för att få kvalitet på programvarorna och hur viktigt detta är. Det är så lätt att bli hemmablind. Man tänker inte på att det är först när man arbetar med en programvara som man upptäcker svagheter. Efter utvärderingen tror jag vi tänker mer på kvalitetsfrågorna samtidigt som vi blivit mer vaksamma när det gäller att få fram programvara med kvalitet.

Vilka var svårigheterna med utvärderingen? Vad skulle ni vilja ha gjort annorlunda?

Samtliga representanter pekar på svårigheterna att på kort tid förklara för andra i testgruppen hur programmet fungerar. Ett flertal av de personer som ingår i utvärderingsgruppen vet i början praktiskt taget ingenting om programmet som ska utvärderas. På ett par dagar ska dessa sätta sig in i programmet för att sedan tillsammans med dem som kanske varit med om att utveckla programmet utarbeta nödvändiga programattribut och sätta kravnivåer för de olika egenskaperna.

– Vi hade en rätt så liten produkt för utvärdering så jag tycker ändå vi lyckades ganska bra med att förklara den i stora drag. Man kan dock inte ge full insikt på den korta tid som stod till förfogande och det märks delvis på resultaten. Om man inte kan förklara produkten kan det ge olika resultat, vilket jag tycker det fanns några exempel på säger Ola Andersson på ZoomON.

Niklas Jardeby på Inobiz har liknande erfarenheter:

– Huvudproblemet för oss var att i och med att vår programvara är så specialiserad är det svårt att lära ut den på så kort tid. Vi har erfarna användare inom den här branschen samtidigt som vi har två dagars utbildning innan vi släpper programvaran till våra kunder. Ingen programvara säljs utan denna utbildning. Det är inte meningen att man ska kunna plocka program-

varan från hyllan och börja använda den. I grund och botten arbetar programmet med ett programmeringsspråk som kräver vissa förkunskaper.

– När man inte kan tränga in i programvarans funktion på djupet är risken att man vid utvärderingen tittar mer på utseende och användarvänlighet än själva kvaliteten.

Henrik Danielsson, Benefit:

– Ja, det är lätt hänt att kraven blir lite väl generella. Man vet inte om man ska titta på kundens användare eller användare generellt. Ändå tycker jag vi hann ganska långt på två dagar. Fast helst skulle man ju behöva några veckor för utvärderingen.

Ovanstående kan vara värt att kommentera. Ytterst handlar de två dagarna före själva utvärderingen om kunskapsöverföring, där de som kan programmet och kanske varit med om att utveckla det, ska förklara för andra personer som inte vet mycket om just detta programmet. Det är ingen lätt uppgift, men det förutsätter nog en hel del förberedelser och helst också pedagogiska kunskaper. Problemet är ju likartat när det gäller att förklara programvarans funktion och användningsmöjligheter för köpare och användare.

Kanske är det nödvändigt för många att lägga ner mer krut på att förklara en produkts programvaras funktion?

– Ja, kommenterar Ola Andersson, efter själva testerna borde man gå igenom resultaten och fokusera på hur funktionerna är beskrivna.

En annan uppgift som tog en hel del tid i anspråk var naturligtvis utarbetandet av produktattribut och kravnivåer. Här stöter man på flera svårigheter samtidigt som just diskussionerna i grupp mellan personer med olika erfarenheter ger värdefulla resultat.

En person som är van vid den aktuella programvaran tycker att den betar sig på ett förväntat sätt. En utomstående kanske däremot förväntar sig något annat och tycker att programvaran utför sin uppgift på ett mindre bra sätt. Sådana åsikts- och erfarenhetsskillnader kan ge upphov till diskussioner.

– Om man själv kan programmet är det lätt att bli partisk och då formulerar man krav som man utan svårighet kan klara av. I ett sådant läge måste utvärderingsledarna rycka in och sätta krav som de tycker man bör klara av. Därför är det mycket viktigt att det i utvärderingsgruppen finns med utomstående som kan titta på detta, även om det kan vara svårt att få fram budskapet om hur programvaran fungerar. Utomstående ”tvingar” oss att sätta högre krav än man annars skulle gjort och det leder naturligtvis till att funktion förbättras och framför allt kvalitet höjs, säger Niklas Jardeby och tillägger:

– Det är absolut nödvändigt att någon oberoende person leder utvärderingen samtidigt som det ställs mycket höga krav på honom eller henne. Bäst vore om man hade en person som både kan ISO-standarden och har lärt sig tillräckligt om den programvara som ska testas.

Hur har utvärderingen påverkat er syn på programutveckling, kvalitet och kundbehov?

– Jag tror att vi i framtiden kommer att lägga större vikt vid kvalitet och därmed också arbete mer arbete med produktifiering. Det är inte bara en programvara som tas fram, utan en produkt. Programvarorna kommer att bli mera mogna tack vare egendeklarationen, tror Henrik Danielsson och fortsätter:

– Utvärderingen har också gett oss en större insikt om standard. Mer standard kan ge bättre funktioner och dessutom besparingar. Detta kan vara en konflikt för vissa som kanske föredrar att gå sin egen väg. Standarden kan naturligtvis också utgöra en viss risk om man fastnar i det gamla.

Niklas Jardeby på Inobiz återkommer till att utvärderingen minskar på hemmablindheten.

– Vi har fått synpunkter på programvaran och dess funktion som vi kanske inte skulle kommit på själva. Erfarenheterna från utvärderingen har gjort att vi lagt ner mer tid på anpassning till Windows-standarden, även om förändringarna inte är radikalt annorlunda. De flesta ändringarna som gjorts efter utvärderingen är dock mest av kosmetisk art om man i detta också räknar in vissa kortfunktioner.

Ola Andersson på ZoomON tycker att utvärderingen direkt påverkat synen på utveckling av programvaror.

– Vi som varit med under utvärderingen börjar nog tänka mer på hur användaren ser på programmen. Tidigare såg man det hela i första hand som en utveckling i sig. Samtidigt tror jag det vore värdefullt om man i framtiden kunde engagera flera från det egna företaget i utvärderingen så att dessa tankar kunde genomsyra företaget. Annars finns det kanske risk att man tappar bort en del av den kunskap man förvärvat.

Vad kan utvärdering och certifiering betyda för programvaruföretagen och för branschen som helhet?

Henrik Danielsson:

– Certifiering tror jag framför allt är bra ur marknadsföringssynpunkt.

När kunderna förstår innebörden av certifieringen, då blir den väldigt värdefull. Jag hoppas detta blir en standard som också tas upp av fristående institut, t ex SEMKO så att det hela drivs vidare på bred front. Då skulle det i bästa fall kunna bli lika betydelsefullt som TCO-märkning av bildskärmar.

Niklas Jardeby, Inobiz:

– Om certifieringen blir allmän och betydelsefull blir den mest viktig för mindre företag. De kan då tävla på mer lika villkor än större företag med en kvalitetssäkring i botten. Sen beror det förstås också mycket på vad som ingår i en utvärdering och vad man får godkänt.

På ZoomON understryker man att för deras del, med 80% av marknaden utomlands, är det viktigt att certifieringen får internationellt genomslag. Då ökar dess betydelse avsevärt.

5. Möjligheter inför framtiden

Föregående kapitel slutade med en förhoppning om att tillämpning av den egendeklaration och certifiering som vi nu ser början på inte begränsas till Sverige utan också kan få internationellt genomslag. Det är en rimlig förhoppning som på sikt skulle kunna bli verklighet och dessutom utvecklas ytterligare. Inspiration och vägledning kan hämtas från flera håll. Några sådana aspekter kommer att beröras i det följande.

TCO:s märkning av bildskärmar började i liten skala i Sverige men är idag accepterad och spridd runt om i världen – mot många odds, skulle man kunna säga.

Utvärdering och certifiering av programvaror enligt den svenska metoden skulle mycket väl kunna spridas till andra länder. Ju större spridning den får, desto värdefullare blir den och desto större blir nyttan för de företag som tillämpar denna metod. De största vinnarna blir förhoppningsvis ändå köparna och användarna, som får kvalitetssäkrade programvaror. Och köparna finns inte bara i Sverige utan än mer utanför vårt lands gränser.

Kvalitetssäkring i andra länder?

Såvitt författaren vet är den svenska utvärderingsmetod för egendeklaration som här presenterats den enda i sitt slag i världen. Därmed ska man inte förneka att man i andra länder arbetar på olika sätt med att försöka testa och utvärdera programvaror. Problemet med att få ett mått på programvarors kvalitet och funktion är universellt och det finns många goda krafter som verkar i den riktningen. Det faktum att vi redan har etablerade internationella standarder, som utgör ramverk för utvärdering av programvaror, är i sig ett stort framsteg.

I det här avslutande kapitlet kan det därför vara lämpligt att gå utanför Sveriges gränser och ge några exempel från andra länder på arbeten som rör programvarors kvalitet. Det bör betonas att det handlar om exempel och inte någon översikt. Vad som är intressant i sammanhanget är snarast att peka på gemensamma problem och sätten de behandlas på i olika kulturer.

Problemen handlar inte bara om att ta fram praktiskt användbara metoder för själva utvärderingen. Utomlands pågår också diskussioner om vem som ytterst bör ansvara för utvärderingsmetoderna, vem som ska utföra utvärdering och certifiering och vem eller vilka som ska stå som garant för de resultat som man får fram vid utvärderingen.

Den svenska metoden i sin nuvarande form är ett självreglerande system som är utvecklat av programvarubranschen. Utvärderingen görs genom självdeklaration av berörda företag, eventuellt med hjälp av inkallade experter. På lite sikt kan utvärderingen göras av fristående institut som t ex SEMKO eller andra, liknande testorganisationer. Man kan naturligtvis också tänka sig att statliga organ skulle kunna stå för certifieringen.

Konferenser om certifiering

Frågor av det här slaget diskuteras på internationell nivå. Under 1999 hölls konferensen Issuing Software Certificates of Quality, med beteckningen ISACC '99, då sådana frågor diskuterades. En ny konferens, ISACC 2000, kommer att hållas i slutet av år 2000. Då är det också tänkt att ta upp frågor som rör certifiering, certifieringsetik, statens roll när det gäller mjukvarukvalitet m m. Konferensen har samma namn som tidigare, dvs Issuing Software Certificates of Quality. I förkortad form heter den ISACC 2000.

Det finns alltså internationella, etablerade fora för den här typen av frågor och det är naturligtvis önskvärt att Sverige håller sig framme i sådana här sammanhang eftersom vi bör kunna tillföra en del erfarenheter i ämnet.

EUREKA

I Europa har man inom EUREKA-projektet "Software Components for the Industry" (Ref. EU1135) uppmärksammat problemet med rätt kvalitet för programvaror, särskilt inom IT-området. I nämnda projektet har man arbetat med att försöka förbättra produktiviteten och kvaliteten av mjukvaruprodukterna genom att försöka få fram en slags komponentpolicy med syfte att återanvända befintliga programvaror eller delar därav. Detta skulle kunna underlättas med hjälp av olika tekniker och verktyg som stödjer registrering och underhåll av programvaror, sökning av mjukvaror eller delar därav så att de lättare kan användas i nya tillämpningar.

En komponentpolicy förutsätter att man kommer överens om en be-

stämd utvecklingsprocess vid framtagning av programvaror. I detta ingår att man bryter ner utvecklingsprocessen i separata faser som står i ett bestämt förhållande till varandra så att de kan testas på ett konsekvent sätt.

Ett annat spår inom samma projekt har gällt utvärdering och certifiering av programvarukomponenter. Det senare har ansetts som en nyckelfråga för att kunna få fram en komponentpolicy vid framställning av mjukvaror.

ESPRIT

Inom ramen för ESPRIT 4-programmet har ett annat projekt med inriktning på mjukvarukvalitet bedrivits. Det är ”Software product advanced certification and evaluation – user focus” (Akronym SPACE-UFO, (Ref nr 33803)).

Huvudsyftet i detta projekt har gällt utveckling av en metod för specifikation och utvärdering av programvarukvalitet med betoning på användarens behov. Metoden bygger på strukturerade intervjuer med användare och leder fram till en systematisk beskrivning av den omgivning som programvaran ska användas i. Användningen av metoden kan bli en hjälp till att göra det lättare för användare och utvecklare av programvaror att bättre komma överens om vad man menar med mjukvarukvalitet. I förlängningen har man inom projektet också arbetat på att få acceptans från IT-branschen på metoden som en slags de-facto-standard.

Dansk metod enligt ISO/IEC

I det tidigare ESPRIT 2-programmet har också utvärdering av programvaror studerats genom, bl a i projektet SCOPE. Förkortningen står för Software CertificatiOn Programme in Europe och pågick under tiden 1989 till 1993. Baserat på detta arbete har senare ett dansk företag, Delta Danish Electronics, utvecklat en metod för utvärdering av programvarukvalitet. Metoden, med beteckningen MicroScope, har baserats på ISO/IEC 9126 och ISO/IEC 14598-5, dvs samma standarder som den svenska metoden bygger på.

Utvärderingen omfattar sex huvudegenskaper: funktion, tillförlitlighet, användbarhet, produktivitet, underhållsmässighet och flyttbarhet, dvs samma som ingår i den svenska metoden. Däremot tycks man inte tagit i beaktande den sjunde huvudegenskaper, nämligen användarkvalitet.

MicroScope-metoden bygger inte på egendeklaration, utan det är före-

taget som utför värderingen. För en inbäddad tillämpning räknar man med en insats på 5-10 persondagar, för en medelstor PC-tillämpning 10-25 persondagar och för en stordatorbaserad tillämpning för industriella processer, ekonomiska transaktioner etc räknar man med en tidsinsats på mer än 25 persondagar .

Företaget har vid det här laget utfört ett 80-tal utvärderingar och har av danska myndigheter blivit ackrediterad att använda metoden i överensstämmelse med EN 45001. Det är en CEN/CENELEC standard för "General criteria for the operation of testing laboratories". Organisationer i Ungern och Grekland har av Delta licens att utvärdera programvaror enligt MicroScope-metoden.

Privatcertifiering i USA

I USA finns det flera exempel på företag som på olika grunder certifierar programvaror. Underwriters Laboratories Inc. marknadsför en certifieringsmetod, UL 1998, som bygger på Standard for Software in Programmable Components. Standarden utvecklades med underlag från industri och berörda myndigheter.

Certifieringen bygger på att företaget går igenom programvaran ifråga för att kontrollera att den uppfyller de krav som anges i de standarder man hänvisar till, huvudsakligen med hänsyn till säkerhet.

Ett annat amerikanskt företag, Reliable Software Technologies, förespråkar en certifieringsmetod som är knuten till den miljö där programvaran ska användas i. Varje certifieringsprocess ska utgöra en funktion av programvarans egentliga "arbetsprofil", däremot inte den profil som marknadsföraren av programvaran förespråkar. I metoden ingår bl a att man försöker upptäcka mjukvarurelaterade anomalier som i värsta fall skulle kunna orsaka katastrofer om de inte rättas till.

En programvara kan exempelvis bete sig inkorrekt med hänsyn taget till specifikationen utan att det har någon negativ effekt på systemet. I ett annat fall kan förhållandet vara omvänt. Programvaran kan bete sig korrekt enligt kraven men kan ändå störa systemet den arbetar i som helhet.

Ett sådant exempel utgör fallet med Arianeraketerna 4 och 5. Till att börja användes en viss programvara i Ariane 4-raketen. Programvaran visade inge felaktigheter och raketen uppförde sig som förväntat. Senare överfördes samma programvara till Ariane 5. Då fungerade inte programvaran alls som man tänkt sig och en katastrof inträffade. Detta hade att

göra med att programvaran betedde sig annorlunda i den miljö som fanns Ariane 5-raketen, bl a till följd av att en del av programvarans beteende i den äldre raketen överfördes till den nyare raketen.

Med den förespråkade metoden ser man vid testning programvaran som en svart låda (black box), som sedan matas med olika indata utifrån programvarans arbetsprofil. Profilen är inte baserad på programvaran, utan på systemets krav.

Metoden används i första hand på programvaror i kategorin COTS, COmmercial of The Shelves.

Framgångsfaktorer

Därmed lämnar vi tills vidare den utländska utflykten bakom oss och återgår till den svenska metoden för utvärdering av programvaror. När den nu lanserats kan det vara intressant att också blicka framåt. Det kan vara lämpligt att fråga vilka faktorer som kan vara avgörande för om metoden kommer att spridas och bli en framgång inom och helst också utom Sverige. Vidare kan det vara lämpligt att fundera på hur själva metoden i sig kan utvecklas på lite sikt. Ty ett är säkert. På en marknad som är så expansiv och dynamisk som programvarubranschen, kan en metod i dess tjänst bara överleva om den utvecklas i motsvarande grad.

När man studerar förutsättningarna för framgång ligger det nära till hands att titta på hur det tidigare nämnda TCO-märket för bildskärmar nådde sin framgång. Några av de faktorer som ligger bakom framgången är bl a följande.

Först och främst fanns det en drivande kraft och entreprenör bakom projektet, Per Erik Boive på TCO:s utvecklingsenhet.

Man hade kanske väntat sig att en fackföreningsman skulle gå den vanliga vägen via remisser och utredningar. Det gjorde han också till att börja med, men insåg så småningom att det var en alltför trög och långsam väg för att han och TCO skulle kunna uppnå resultat inom rimlig tid.

Istället tog han direkt kontakt med tillverkarna av bildskärmar och försökte övertyga dem om att de borde arbeta för en gemensam standard som skulle ge säkrare bildskärmar. En sådan standard skulle gynna både tillverkarna och användarna.

Kanske kan ett sådant agerande tyckas vara entusiastiskt i överkant om man betänker att flera av dessa bildskärmstillverkare var mycket stora. I kraft av sin storlek var flera av dem vana att själva sätta sina egna standar-

der. Trots detta var de ändå påverkbara och lyssnade uppenbarligen till de argument som fördes fram.

Bundsförvanter

Utöver detta skaffade sig Boive också bundsförvanter genom Naturskyddsföreningen och NUTEK. Därtill ska man naturligtvis inte glömma det nätverk som han byggde upp och som är nödvändigt för att sprida nya idéer till så många som möjligt.

Slutligen kommer man tillbaka till entreprenören, visionären som genom sitt nätverk och till de olika organisationer han engagerade, ständigt kunde förmedla sin vision, hålla den levande och utveckla den vidare efterhand som den började förverkligas.

I bildlig bemärkelse handlar ett sådan här projekt om att försätta berg. I detta ingår att man måste få till stånd en kritisk massa, dvs få tillräckligt många personer, företag och organisationer att ta till sig det nya konceptet och att börja tillämpa det. När några har kommit igång med tillämpningen är det allt fler som följer efter därför, antingen för att de tror på idén eller möjligen för att de inte vågar stå utanför.

Vidareutveckling

Om vi återgår till metoden för egendeklaration av programvaror och vad det står för, är det alltså viktigt att man genom visions- och missionsarbete ständigt för ut budskapet om det nya konceptet. Ju fler företag som börjar egendeklarera sina programvaror, desto större betydelse får metoden och därmed ökar också möjligheterna att efterhand utveckla konceptet ytterligare.

Vidareutveckling är också en förutsättning för fortsatt framgång.

När det gäller kvalitetsmärkning av programvaror enligt SPI:s metod så kan man tänka sig en utveckling på flera sätt.

För det första kan man räkna med att branschen efter hand självt kommer att öka kraven så att utvärderingen, dvs man ställer upp allt hårdare krav på programvarornas funktion och kvalitet. Kanske utvecklas metoden och kraven så att det motiverar ett nytt märke ungefär som att TCO-märket numera finns både som TCO-92, TCO-95 och TCO-99.

För det andra är det möjligt att branschen efter hand utvecklar produktspecifika regler för programvaror som kan placeras i samma kategori.

Detta är av intresse om det visar sig att vissa programvaror har så pass mycket likheter i funktion och egenskaper att man vill göra mer detaljerade jämförelse än vad som är möjligt med de befintliga, generella reglerna. När ett sådant läge inträder kan branschen ta fram produktspecifika regler som underlättar jämförelsen på större detaljnivå än som är möjligt med de generella reglerna. Formerna för hur sådana produktspecifika regler ska tas fram finns redan förberedda.

Förslag till produktgruppspecifika utgångspunkter kan efterhand utarbetas i samarbete mellan intresserade programföretag/distributörer samt bransch- och intresseorganisationer. ITS – Informationstekniska Standardiseringen kan vid behov bistå i detta arbete.

För det tredje kan man tänka sig att fristående institut, t ex SEMKO, bygger upp en kompetens som gör att de som tredje part kan åta sig att utvärdera programvaror. Därmed betonas objektiviteten ytterligare.

För det fjärde borde det inte vara omöjligt att också sprida metoden och dess utvecklingsmöjligheter utomlands. Det faktum att den är baserad på internationella standarder utgör en bra plattform. Om vi i Sverige dessutom kan visa att metoden fungerar i praktiken och att såväl användare som utvecklare och leverantörer av programvarorna har nytta av utvärderingsresultaten borde metoden kunna exporteras både inom och utom Europa. Det kräver förstås också en hel del marknadsföring och att man håller visionen levande.

Efterfrågade & aktuella TELDOK-rapporter om erfarenheter av tidig IT-användning

Utkomna rapporter från 1994 och senare (ca 80 st) kan beställas – gratis i enstaka ex – från <http://www.teldok.org>, från Lindegården tel 020 23 00 11 eller från teldok@ett.se. Ange rapportnummer för säker leverans!

Den som i fortsättningen önskar erhålla skrifter från TELDOK får för närvarande automatiskt alla TELDOK Rapport och alla TELDOK-Info, löpande, gratis, i ett ex. Anmäl intresse för detta, liksom adressändringar och liknande, till Anna Karlstedt, FAX: 08-32 65 24, eller till teldok@ett.se.

Nedan förtecknas de fem senaste rapporterna samt de fem rapporter som ”beställts mest” under 2000. Trevlig läsning önskas! Läs mer om nyare och gamla rapporter (eller läs 45 av rapporterna i elektronisk version) på...

<http://www.teldok.org>

De 5 senaste rapporterna

TELDOK Rapport 135

IT, innovation – Israel

Bengt-Arne Vedin (red)

Israel har fler IT-företag noterade på den amerikanska teknikkörsen NASDAQ än hela Europa tillsammans. Rapporten dokumenterar en studieresa till Israel i januari 2000, med besök på IT-företag, riskkapitalföretag samt ”inkubatorer” och industriparker för nyföretagande.

TELDOK Rapport 134

Privatliv & Internet – som olja och vatten?

Anders R Olsson

Behandlar problemen med att skydda den personliga integriteten i IT-samhället; beskriver riskerna och diskuterar möjligheterna att värna privatlivet med juridiska, tekniska och praktiska metoder.

TELDOK Rapport 133

Interaktiv underhållning inför framtiden

Erik Fjellman & Jan Sjögren

Rapporten syftar till att beskriva och förklara vad dataspel och dataspelande innebär, och hur den interaktiva underhållningen kommer att förändras och påverka oss i framtiden. Speltyper och spelgenrer, användning och användare, teknologin bakom spelen, dataspelsbranschen och våldsdebatten behandlas.

TELDOK Rapport 132

Den digitala fabriken – Verkstadsföretaget som IT-företag

Christina Johannesson & Peter Kempinsky

Rapporten baseras på intervjuer med fyra verkstadsföretag och deras samverkanspartners. Fallstudierna har olika inriktning och profil och uppvisar samtliga en innovativ och kvalificerad användning av IT som stöd för tillväxt och samverkan.

TELDOK-Info 18

Innovation & Internet

Bengt-Arne Vedin

Hur kan vi organisera, förstå och diskutera den mångfald av tjänster och ”produkter” – innovationer – som är relaterade till Internet? Exemplet gäller nya typer av aktörer, nya system för upphandling och prissättning, att värva kunder redan innan man har något att erbjuda.

De 5 mest beställda rapporterna

TELDOK Rapport 131

IT-Sverige – en vision från Dataföreningen

Bengt-Arne Vedin

Dataföreningen (www.dfs.se) tog initiativ till rapporten för att fira sitt femtioårsjubileum med ”ett antal frambildsbilder”, en Delfi-studie, ett antal scenarier, tillsammans alltså ”antal studier av utvecklingsläge och framtidstendenser”. Rapporten ”lovade inte att framtiden kommer att bli på något speciellt sätt...”; men att ta del av förtätade framtidsbeskrivningar är ett bra sätt att förbereda sig.

TELDOK Rapport 132

Den digitala fabriken – Verkstadsföretaget som IT-företag

(se ovan)

TELDOK Rapport 130

TELDOKs Årsbok 2000

Red: Gull-May Holst

Årsboken behandlar de våldsamma förändringarna i IT- och telebranscherna, Internet-användningen liksom användningen av IT och tele, i Sverige och i övriga världen, liksom uppgifter om några centrala användningsområden för IT. Årsboken är rikt illustrerad och innehåller fascinerande ”tidsfönster” som visar IT-utvecklingen och andra minnesvärda utvecklingssteg bakåt i tiden.

TELDOK Info 18

Innovation & Internet

Bengt-Arne Vedin

(se ovan)

TELDOK Rapport 125

IT i framtidens lärande

Lars Bolander

Fram till idag har lärandet varit förbehållet skolor, skolböcker och lärare. IT kan förändra detta helt. Ett antal konkreta praktikfall med exempel på ”rader av nya grepp kring lärandet med stöd av IT” demonstreras i rapporten, både inhemska och utländska.



Egendeklaration av programvaror

När inte programmerarna kan anpassa sina program till användarna, kostar det näringslivet miljarder i väntetid och konsultarvoden. Därför har en svensk världsunik varudeklaration för programvara – Spi2000 – skapats och prövats. Via TELDOK 39: **Egendeklaration av programvaror** beskriver hur de första utvärderingarna gick till på den svenska marknaden och vilka erfarenheterna som skapades.

Författare till **Egendeklaration av programvaror** är konsulten och journalisten Bertil Håkansson (bertil.hakansson@infocomab.se), med inriktning på teknik och vetenskap vid InfoCom AB.

TELDOK och KFB ger ut rapporter i programmet **Telematik 2004** för att ”studier av tidiga användare kan vägleda beslut och åtgärder inför framtiden”.

TELDOK Rapport

134: **Privatliv & Internet – som olja och vatten?** behandlar problemen med att skydda den personliga integriteten i IT-samhället, beskriver riskerna och diskuterar möjligheterna att värna privatlivet.



TELDOK Rapport 133: **Interaktiv underhållning inför framtiden**

beskriver och förklarar dataspel och dataspelande, och hur den interaktiva underhållningen kommer att förändras och påverka oss i framtiden.



Hur uppstår ”nya idéer som lyckas” – innovationer – på, för, tack vare Internet? Och hur driver man affärer kring innovationerna på webben?



TELDOK-Info 18: **Innovation & Internet** beskriver innovationer på (och för) Internet. Produkterna och tjänsterna sätts in i ett större sammanhang, och rapporten ger råd och tips.

TELDOK driver också webbsajten **www.teldok.org/innova** där det finns (och tillkommer) ännu mer information.

Rapport 135: **IT, innovation – Israel** dokumenterar en studieresa till Israel januari 2000. Fram träder ett myller av livaktiga IT-företag (webb-pionjärer, e-handelsföretag, sådana som sysslar med mobilt Internet), israeliska och amerikanska riskkapitalister, statliga och privata ”inkubatorer” och industriparker för nyföretagande.



Via **teldok** • ISSN 0283-5266 • 2000 • Pris 100 kr

Information: <http://www.teldok.org/>

Beställ gratis rapporter från Turess lada: 020 23 00 11

Läs mer om aktuella & efterfrågade publikationer längst bak i rapporten!

www.teldok.org