

## Lingvister – bygger broar mellan programmerare och programanvändare

---

Sara Nyholm  
Språktjänsten  
Vasa universitet

*Erilaiset tietojärjestelmät ovat nykyään iso osa arkipäivää ja työelämää. Yhä enemmän käytetään teknisiä ratkaisuja eri tarpeisiin. Jotta käyttäjä oppisi käyttämään kaikkia järjestelmiä tarkoituksenmukaisella tavalla, on tärkeää että järjestelmien käytettävyys on hyvällä tasolla. Käyttäjien pitäisi helposti ymmärtää miten järjestelmät toimivat, jotta he voivat käyttää niitä tehokkaalla tavalla. Käytettävyydessä myös kieli on merkittävä osatekijä. Toimiva terminologia auttaa käyttäjiä ymmärtämään miten järjestelmä toimii. Artikkelissani esittelen tapaustutkimuksen jonka tein suomalaisessa ohjelmistoyrityksessä vuonna 2014. Tutkimuksessa nostan esiin ongelmallisia lokalisoiteja ja niiden ratkaisuprosessin. Tutkimuksen tavoite on kehittää malli, jonka avulla voidaan nostaa kielellistä käytettävyyttä tietojärjestelmissä.*

**Nyckelord:** datasystem, lokalisering, språklig användbarhet

### 1 Användbarhet

I takt med att ny teknik utvecklas får vi nya program och system att använda, både i det privata och i det professionella livet. I och med att datasystemen blir fler och vanligare ställer det högre krav på användarna att lära sig nya saker. Samtidigt ökar kraven på systemens användbarhet – det ska vara enkelt att lära sig de nya programmen. Användbarhet kan definieras som ett mått på hur bra användarna kan använda en bestämd produkt för att nå bestämda mål så att de ger resultat på ett effektivt sätt (ISO 9241-11 1998). ISO-standardens användbarhetsdefinition har dessutom utökats av Nielsen (2001) som menar att användbarhet också innefattar ett mått på hur lätt det är att lära sig ett system, hur lätt man kan komma ihåg hur det fungerar samt att systemet ska innehålla så få fel som möjligt. Donahue, Weinschenk och Nowicki (1999: 3) och Molich (2006: 23) förklarar användbarhet som ett mått på hur bra användarna kan lära sig att använda ett datasystem, hur systemet hjälper användarna att komma ihåg vad de lärt sig, hur risken för fel reduceras, hur effektiva användarna kan vara och hur ändamålsenligt systemet fungerar.

Keränen, Lamberg och Penttinen (2005: 48) säger att användbarhet är måttet på datasystemets kvalité när man bedömer systemet ur användarens synvinkel. Det är alltså användaren som är i fokus i användbarheten. Det bekräftar också Krug (2006: 11) när han på frågan om vad som är den viktigaste faktorn i användbarheten svarar ”don’t make me think!”. Med det menar han att det ska vara uppenbart för användarna vad de ska göra utan större ansträngning. Fokus ska således alltid ligga på användarna.

Språket är en del av användbarheten i alla datasystem. Systemen innehåller olika *lokaliseringar*, som hjälper användarna att förstå vad de ska göra. Det kan vara menyer, olika dialogfönster och felmeddelanden (Esselink 2000: 57; Schmitz 2014: 452). Dessa ska vara formulerade så att användarna förstår vad de förväntas göra eller vad som händer när de klickar på en knapp eller en länk. Nielsen (2001: 101) och Molich (2006: 116) säger att det är viktigt att uttrycka sig kort men precist när man skriver på webben, vilket betyder att lokaliseringarna ska vara korta och uttömmande. Med tanke på att den primära riktlinjen för användbarhet länge har varit att tala användarens språk (Nielsen & Budiu 2013: 111) sätter detta en stor press på att de som utformar lokaliseringarna ska kunna producera ett språk som är korrekt och ändamålsenligt med tanke på målgruppen. För att den språkliga användbarheten ska vara hög i ett program bör även lokaliseringarna uppfylla de faktorer som användbarhet är uppbyggd av. Lokaliseringarna ska:

1. vara lätta att förstå
2. underlätta minnet så användarna kommer ihåg hur det fungerar
3. vara effektiva
4. vara ändamålsenlig
5. reducera och minimera risken för fel

Lokaliseringar som består av begrepp med snarlika synonymer kan vara problematiska för läsarna i och med att en variation av termer för samma begrepp kan orsaka missförstånd (jfr Nissilä & Nuopponen 2012, 2013), men det kan också fungera andra vägen, dvs. att snarlika begrepp beskrivs med samma term. Därför är det viktigt att inte se på en lokalisering åt gången utan se lokaliseringen i sin kontext i programmet när man utformar lokaliseringar med en så hög användbarhet som möjligt.

I utformningen av lokaliseringar med hög användbarhet kan man dra paralleller till terminologin. Nuopponen och Pilke (2010: 65) säger att en god term ska bl.a. vara kort, beskriva begreppet, skilja sig från andra termer och passa in i begreppssystemet. De teoretiska principerna för en god term kan dock vara svåra att uppfylla i praktiken. Där behövs det istället någon form av kompromiss. (Nuopponen & Pilke 2010: 65.) Lokaliseringar behöver innehålla samma element som termer för att fungera väl, men även här är det teoretiska idealet svårt att uppnå i alla situationer.

Teknisk dokumentation har undersökts tidigare och framför allt när det gäller längre instruktioner om en funktion eller ett program (se Spinuzzi & Zachry 2000; Virtaluoto 2013). Även forskning i terminologiskt lokaliseringsarbete har bedrivits, bl.a. av Schmitz (2005, 2014), men det finns fortfarande behov av forskning om språkets roll i användbarhet och om hur språket ska beaktas i datasystem eftersom databranschen utvecklas ständigt. För att öka förståelsen har jag gjort en fallstudie av ett antal lokaliseringar vid ett finländskt dataföretag och analyserat hur problematiska lokaliseringar

har hanterats från identifiering av ett problem till lösning. Studien är praktiskt orienterad för att ge handfasta riktlinjer för utvecklingen av lokaliseringar med hög användbarhet. Företaget specialiserar sig på program för skolor och undersökningen är gjord under året 2014.

### 1.1 Syfte, material och metod

Syftet med min fallstudie är att genom en praktisk studie av problematiska lokaliseringar i ett datasystem presentera en modell för hur lingvistiskt arbete kan höja den språkliga användbarheten i ett datasystem. Jag gör det genom att presentera och diskutera hur problemlösningsprocessen går till på dataföretaget.

Det undersökta företaget utvecklar dataprogram för finländska läroanstalter. Programmet riktar sig till olika skolstadier och används av såväl personal som av eleverna och deras vårdnadshavare. Det betyder att lokaliseringarna utformas för att passa en brokig användargrupp och görs dessutom på svenska, finska och engelska. På företaget jobbar sju lingvister. Fyra ansvarar för de finska texterna, två för de svenska och en för de engelska.

I det praktiska arbetet med lokaliseringarna bygger programmerarna upp och utvecklar funktionerna och gör samtidigt de första utkasterna till lokaliseringarna. När funktionen är klar ur programmerarnas synvinkel går lingvisterna igenom funktionerna för att kontrollera och översätta lokaliseringarna. Ansvar för både opublicerade och publicerade lokaliseringar ligger på lingvisterna. Det betyder att lingvisterna kontrollerar opublicerade lokaliseringar, men också tar emot användarresponsen som gäller lokaliseringar som redan finns tillgängliga i programmet.

I min studie har jag undersökt fyra svenska lokaliseringar som på ett eller annat sätt har orsakat problem eller missförstånd. De kan därmed ge mer information om hur problemlösningen går till. Lokaliseringarna är följande:

1. man/kvinna
2. studerande
3. resurs
4. kalendernoteringarna hämtas för perioden som börjar x veckor efter denna vecka och slutar x veckor efter denna vecka

Lokalisering 1, 3 och 4 påverkar endast ett ställe i programmet. Den andra lokaliseringen innehåller ordet *studerande* som används i 263 lokaliseringar. Det betyder att om man ändrade *studerande* skulle man behöva göra det på 263 ställen. Nämnas bör också att lokalisering 1, 2 och 3 är sådana som har blivit publicerade och som användarna haft

tillgång till, medan lokalisering 4 aldrig blivit publicerad. Lokaliseringarna 1, 2 och 3 är till synes inte i sig begränsade till ett datasystem utan kan användas i många olika kontexter. Orsaken till att de ändå är av vikt för denna studie är att lokaliseringarna till stor del består av ord och termer som används i allmänspråket, men problemlösningen är begränsad till datasystemets ramar.

## 1.2 Problem, diskussion, användarrespons och lösning

När en lokalisering upplevs som problematisk är det någon av de fem faktorerna i användbarhetsdefinitionen som lingvisterna vid företaget tar fasta på (se ovan). Det kan vara att lokaliseringen är svår att förstå, svår att komma ihåg betydelsen för, ineffektiv, inte ändamålsenlig eller riskerar att bli missförstådd. Alla lokaliseringar i den här artikeln uppfyller någon eller några av dessa faktorer. I analysen tar jag fasta på fem olika steg i lokaliseringarnas process från identifiering av problemet till lösning. Stegen har jag identifierat genom att själv vara med i processen vid företaget och genom att delta i diskussionerna om lokaliseringarna. De fem stegen är:

1. identifiera problemet
2. diskutera problemet
3. presentera olika ändringsmöjligheterna
4. beakta användarresponsen
5. besluta hur problemet ska lösas

### 1.2.1 Man/kvinna

Programmet utvecklas för nästan alla skolstadier. I programmet lagras uppgifter om eleverna och personal och där kan man fylla i vilket kön de är av. Åldersspannet bland användarna är mycket stort och kan vara alltifrån små förskolebarn till pensionärer. Det gör att *man* och *kvinna* ibland är korrekt, men ibland borde det vara *flicka* och *pojke*.

Rent tekniskt är det möjligt att ha två olika lokaliseringar för kön, dvs. både *man/kvinna* och *flicka/pojke*, men det kan vara svårt att hitta ett ändamålsenligt sätt att avgränsa användningen. Denna funktion var redan i användning vilket betyder att användarna har haft tillgång till den och lokaliseringen, men inga användare har reagerat på ordvalet *man/kvinna*. Alla former av ändringar medför mer arbete som ger ökade kostnader för företaget och därför är det viktigt att ändringen fyller ett behov. Om användarna inte har det behovet är det ur kostnadssynvinkel rimligare att inte göra någon ändring. Ur användbarhetssynvinkel konstaterade man att trots att lokaliseringen inte är fullständigt ändamålsenlig är funktionen ur användarnas synvinkel effektiv, lätt att lära sig och lätt att använda. Att ändra en funktion eller en lokalisering medför alltid en risk för ökat

missförstånd innan användarna lärt sig de nya rutinerna. Därför togs beslutet att inga ändringar görs.

### 1.2.2 Studerande

Det är olika skolstadier som använder programmen vilket innebär att *studerande* ibland är korrekt, medan *elev* ibland skulle vara att föredra. Lokaliseringen *studerande* är lättare att avgränsa än *man/kvinna* eftersom den är specifik för olika skolstadier och därför går det att göra helt skilda lokaliseringar för t.ex. grundskolor, gymnasier och yrkesskolor där skolorna själva kan välja vilka lokaliseringar de vill använda. *Studerande* är speciell med tanke på att det används i sammanlagt 263 lokaliseringar och en ändring skulle medföra en omfattande komplettering av lokaliseringarna. Användarresponsen för *studerande* visade att endast ett fåtal användare har reagerat på ordvalet.

Ur användbarhetssynvinkel konstaterade man att även om *studerande* i vissa fall är felaktigt så ligger betydelsen så nära *elev* att det inte orsakar missförstånd. Lokaliseringen är lätt att förstå och komma ihåg och fungerar tillräckligt effektivt och ändamålsenligt i och med att endast ett fåtal användare har upplevt den som problematisk. *Studerande* och *elev* står för skilda begrepp, men är ändå snarlika och även om man använder samma term för båda begreppen visar användarresponsen att det går att förstå. Att ändra 263 lokaliseringar kan däremot ge starkt ökad risk för missförstånd. Detta ledde till beslutet att *studerande* får vara kvar i alla lokaliseringar och för alla skolstadier.

Lokaliseringen *studerande* ändrades inte men det finns ändå en alternativ lokalisering. När *studerande* diskuterades lyftes också förskolorna fram. Åsikten som lyftes fram var att det går att tala om *studerande* i grundskolor, men det är inte möjligt i förskolor. Där är det fråga om *barn*. Lokaliseringen *studerande* fungerar inte effektivt och ändamålsenligt i förskolor. Förskolorna hade i detta skede ännu inte tagit programmet i bruk och det fanns ingen användarrespons att tillgå, men man tog ändå beslutet att utveckla skilda lokaliseringar för förskolorna där bl.a. *studerande* är ändrad till *barn*.

### 1.2.3 Resurs

I programmet går det att utforma scheman för studerande, lärare, övrig personal och för olika hjälpmedel. När man gör schemat är studerandena kategoriserade i en grupp, lärarna i en annan och övrig personal och hjälpmedel i en tredje. Den tredje kategorin kallas *Övriga resurser*. Problemet var att övrig personal för det första kallas *resurs* och för det andra att de kategoriseras i samma grupp som resurser, som hjälpmedel. I diskussionen föreslog man att det kunde åtgärdas språkligt genom att byta namn på kategorin, t.ex. till *Övrigt*. Det skulle inte medföra mycket arbete, men samtidigt skulle det inte

heller lösa kategoriseringsproblemet. För att lösa kategoriseringsproblemet skulle man behöva skapa en fjärde kategori och skilja mellan *Resurser* och *Personal*. Det skulle innebära stora ingrepp i källkoden och stora ändringar görs för det mesta inte om det i övrigt är en tekniskt sett fungerande funktion. Användarresponsen visade dock att många användare hade reagerat både på ordvalet och kategoriseringen och därför tog man beslutet att ändra funktionen så att personal och hjälpmedel kategoriseras skilt. Samtidigt eliminerades det språkliga problemet. Även om ändringen medför en ökad arbetsmängd och ökad kostnad valde man alltså att tillgodose användarnas behov för att öka användbarheten.

Lokaliseringen *resurs* gav i sig en felaktig bild av funktionen som den representerade och bidrog därför till att det var svårare att förstå vad den betydde. Den orsakade också viss irritation bland användarna och dessa faktorer ledde till att lokaliseringen inte var lätt att förstå, och var varken effektiv eller ändamålsenlig, dvs. lokaliseringens, men också funktionens användbarhet var inte hög.

#### 1.2.4 Kalendernoteringarna hämtas för perioden som börjar x veckor före denna vecka och slutar x veckor efter denna vecka

I programmet utvecklades en funktion som gör det möjligt att överföra scheman. I överföringen kom följande lokalisering fram: *Kalendernoteringarna hämtas för perioden som börjar x veckor före denna vecka och slutar x veckor efter denna vecka*. Istället för x kunde användaren fylla i hur många av schemats veckor som skulle tas med i länken.

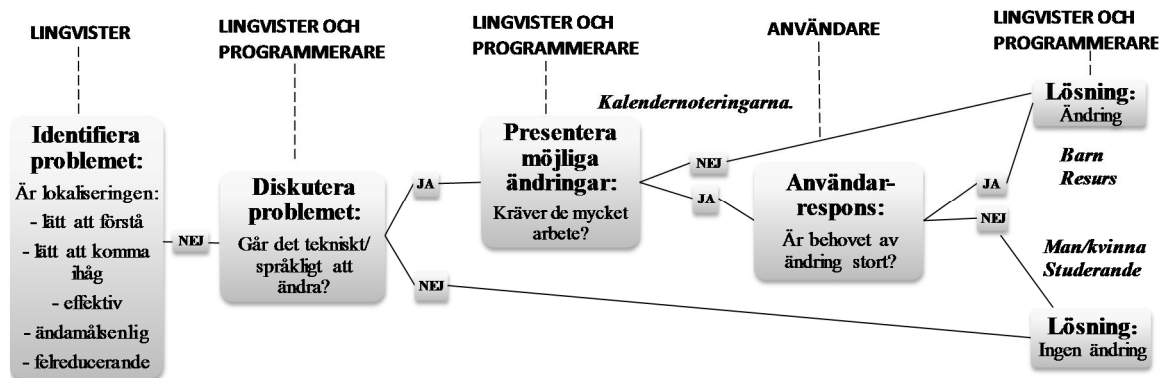
Vid första anblick kan det verka som att det antal veckor som man fyller i är det reella antalet veckor framåt och bakåt som kommer med i kalenderöverföringen. Detta är dock inte fallet. Om man väljer fem veckor bakåt kommer hela den period som pågick för fem veckor sedan att komma med. På motsvarande sätt kommer hela den period som pågår om fem veckor att komma med om man väljer fem veckor framåt. Olika skolor har olika periodsystem, dvs. olika många perioder per läsår och det betyder att antalet veckor som sist och slutligen kommer med kan variera stort mellan skolor och användare, även om de fyller i samma antal veckor. I och med att användarna inte nödvändigtvis känner till skolans periodsystem kan det skapa osäkerhet hos användarna om de upplever att programmet inte tar med rätt antal veckor i överföringen. Det gör att lokaliseringen kan vara svår att lära sig och komma ihåg och är därmed varken effektiv eller ändamålsenlig. Användbarheten är inte hög. För att användarna ska kunna förstå vilka veckor som kommer med behövs en längre instruktionstext, men sådana är inte kutym i programmet. Ur ett användbarhetsperspektiv är det dock viktigt att lokaliseringarna är uppbyggda på samma sätt och att de följer samma kutym. Dessutom är det inte heller ändamålsenligt att ha längre instruktionstexter på webben (Nielsen 2001: 104; Nielsen

& Budiu 2013: 102). Det blev därmed klart att det behövdes mer än en rent språklig ändring.

Lösningen blev att ta bort möjligheten att välja antal veckor så att när användarna överför ett schema får de automatiskt med en tidsperiod som på förhand är bestämd i källkoden. Genom att ändra själva funktionen försvann även det språkliga problemet.

### 1.3 Från problem till lösning

För att få en bättre överblick över hur arbetet med att lösa språkliga problem går till har jag utvecklat en analysmodell som företaget i praktiken använder för att diskutera språkliga problem i programmet. Analysmodellen består av olika frågor och beroende på svaret på frågorna går man vidare i analysen på olika sätt.



Figur 1. Analysmodell – från problem till lösning

Figur 1 visar att det första steget är alltid att lingvisterna identifierar problemet. När de går igenom funktionerna kontrollerar de om lokaliseringarna är lätta att förstå, lätta att komma ihåg, effektiva, ändamålsenliga och felreducerande, dvs. om den språkliga användbarheten är hög. Om en eller flera av dessa faktorer inte uppfylls tar de upp lokaliseringen till diskussion med programmerarna och tillsammans reder de ut om det överhuvudtaget går att lösa, tekniskt och/eller språkligt. Om det inte går att lösa blir det inte någon ändring, men det är mycket sällan som det inte går att hitta någon form av lösning. Därför är svaret på frågan i steg 2 för det mesta *ja*. Följande steg är att presentera och diskutera olika ändringsförslag och deras för- och nackdelar. De olika förslagen kan läggas fram av både lingvister och programmerare. Samtidigt frågar man sig alltid hur mycket arbete ändringarna kräver. Man försöker sträva till att inte göra alltför stora ingrepp varken i källkoden eller i lokaliseringarna för att minimera kostnader och eventuella konsekvenser.

Om ett ändringsförslag är bra och inte kräver mycket arbete leder det till en ändring. Så var fallet med *kalendernoteringarna*. Om en bra lösning kräver mycket arbete går man vidare till nästa steg och ser på vad användarna säger. Om användarna inte har reagerat på lokaliseringen blir det ingen ändring (t.ex. *man/kvinna* och *studerande*). Om användarresponsen visar att många användare har reagerat på lokaliseringen blir det en ändring (t.ex. *resurs*). När det gäller lokaliseringen *barn* fanns ingen användarrespons att tillgå, men man gjorde ändå bedömningen att det krävs skilda lokaliseringar för förskolorna. Alla problem som identifieras går igenom samma process och leder alltid till steg 5: någon form av lösning. Detta gäller även om lösningen innebär att det inte blir någon ändring.

Företaget strävar efter att upprätthålla en hög språklig användbarhet, men ibland behövs kompromisser med tanke på programmets tekniska möjligheter och begränsningar (jfr Nuopponen & Pilke 2010: 65). Kompromisserna görs dock i sådana fall där användarna inte upplever dem som problematiska (t.ex. *man/kvinna*, *studerande*), men i vissa fall är en kompromiss inte möjlig utan det behövs en ändring (t.ex. *barn*, *resurs*). I exemplet med kalendernoteringarna visar det sig också att en funktion som rent tekniskt fungerar ibland ändras för att det språkligt sett inte går att formulera en lokalisering som är ändamålsenlig.

## 2 Konklusion

Fallstudien visar att genom att ta sin utgångspunkt i användbarhetsdefinitionen (se Donahue m.fl. 1999: 3; Keränen m.fl. 2005: 48; Krug 2006: 11; Molich 2006: 23; Nielsen 2001) kan man upprätthålla en hög språklig användbarhet i datasystem. Lokaliseringarna i fallstudien är i sig inte begränsade till datatekniska kontexter, framför allt lokalisering 1, 2 och 3, men det som knyter dem till den tekniska kontexten är att själva problemlösningen är begränsad till datasystemets ramar. Dessutom är de ändringsmöjligheter som finns med alltid direkt kopplade till programmerarnas arbetsmängd och den rådande kutymen i programmen. Det lingvistiska arbetet med lokaliseringarna kan därmed inte utföras separat från arbetet med själva funktionerna.

När ett lokaliseringsproblem uppstår är det framför allt två viktiga aspekter som lingvisterna tillför i identifieringen. För det första har de helhetsbilden över lokaliseringarna i programmen och vet vilka ord och termer som används för de olika begreppen och kan därmed undvika onödig användning av synonymer. Om ett begrepp har flera termer i systemet medför det en ökad risk för missförstånd (jfr Nissilä & Nuopponen 2012, 2013). För det andra har lingvisterna också kunskap om språkets pragmatik och kan därmed utforma fungerande texter och även tillåta kompromisser för att göra den så ändamålsenlig som möjligt. Identifieringen är direkt kopplad till användbarhetsdefinit-



ionen eftersom de frågor som man i praktiken utgår från teoretiskt sett kan anknytas till användbarheten.

I och med att teknikens möjligheter växer och behoven ökar gör det också att data-systemen fortlöpande utvecklas. Källkoden är själva förutsättningen för att ett program ska existera (jfr Esselink 2000: 59) och när det gäller språket i programmet kan det vid första anblick inte kännas lika relevant som koden och man kan ta för givet att en programmerare ska kunna utforma lokaliseringarna. Att i dataföretag anställa personer som har den lingvistiska expertisen kan därmed kännas onödig i och med att det ger en tilläggskostnad. Nielsen (2001: 8) säger att inom programvaruindustrin finns risken att det är viktigare att få fram produkterna i tid snarare än att de grundligt gås igenom innan de publiceras för användarna. Risken med det är att supportavdelningen får en ökad arbetsbelastning. Om programmet omfattande testas, både tekniskt och språkligt, kommer det att spara på supportavdelningens resurser och i förlängningen sänka kostnaderna. Alla former av ändringar medför alltid en ökad arbetsmängd som i praktiken betyder ökade kostnader, men en hög användbarhet kan i längden minska kontakten till supporten och därmed minska arbetsmängden och kostnaderna (jfr Nielsen 2001: 8). Om användbarheten i ett program är låg gör det programmet svårt att använda och det ger en ökad risk för att programmet inte kommer att användas även om funktionerna i sig är värdefulla (Pressman 2005: 631). Hög språklig användbarhet är viktigt för att säkerställa att programmet tas i bruk. Därför kan det i det långa loppet löna sig för företag att anställa lingvister för att effektivisera den språkliga användbarheten i programmen.

All teknisk dokumentation har länge setts som ett stängt system och den språkliga användbarheten har testas skilt från de övriga komponenterna (Spinuzzi & Zachry 2000: 171). Detta synsätt har ifrågasatts och bl.a. Spinuzzi och Zachry (2000: 179–180) har i sina undersökningar lyft fram teknisk dokumentation som ett öppet system, framför allt mellan dokumenterarna och användarna. I företaget som denna fallstudie gäller strävar man efter att utforma lokaliseringar som är enkla att förstå och att komma ihåg, som minimerar risken för fel och som effektiviserar systemet så att det fungerar så ändamålsenligt som möjligt. Fallstudien visar att hög språklig användbarhet blir ett resultat av en kontinuerlig samverkan mellan lingvister, programmerare och användare.

## Källor

- Donahue, George M, Susan Weinschenk & Julie Nowicki (1999). *Usability is good business*. [online] [citerat 27.3.2015]. Tillgänglig: <http://half-tide.net/UsabilityCost-BenefitPaper.pdf>.
- Esselink, Bert (2000). *A Practical Guide to Localization*. Volym 4. Amsterdam/Philadelphia: Johan Benjamins publishing company.
- ISO 9241-11 (1998). *Ergonomic requirements for office work with visual display terminals (VDTs). Part 11: Guidance on usability*. Helsingfors: Finländska standardiseringsförbundet SFS.

- Keränen, Vesa, Niko Lamberg & Jukka Penttinen (2005). *Digitaalinen media*. Borgå: Docendo.
- Krug, Steve (2006). *Don't Make Me Think. Common Sense Approach to Web Usability*. 2:a upplagan. Berkeley: New Riders.
- Molich, Rolf (2006). *Webbdesign med fokus på användbarhet*. Lund: Studentlitteratur.
- Nielsen, Jakob (2001). *Användbar webbdesign*. 1:a upplagan. Stockholm: Liber AB.
- Nielsen, Jakob & Raluca Budiu (2013). *Mobile Usability*. Berkeley: New Riders.
- Nissilä, Niina & Anita Nuopponen (2012). Terminologin näkökulma synonymiaan. I: *Kielet liikkeessä, Språk i rörelse. VAKKI-symposium XXXII*, 219–229. Red. Niina Nissilä & Nestori Siponkoski. Vasa.
- Nissilä, Niina & Anita Nuopponen (2013). Tieteen termit termipankkiin – haasteena synonymia. I: *Perspektivet som utmaning, VAKKI-symposium XXXIII*, 245–256. Red. Maria Eronen & Marinella Rodi-Risberg. Vasa.
- Nuopponen, Anita & Nina Pilke (2010). *Ordning och reda. Terminologilära i teori och praktik*. Norstedts.
- Pressman, Roger S (2005). *Software Engineering. A Practitioner's Approach*. 6:e upplagan. McGraw-Hill Higher Education.
- Schmitz, Klaus-Dirk (2005). Terminological Data Modelling for Software Localization. I: *Terminology and Content Development – TKE 2005, 7<sup>th</sup> International Conference on Terminology and Knowledge Engineering*, 27–35. Red. Bodil Nistrup Madsen & Hanne Erdman Thomsen. Köpenhamn: GTW.
- Schmitz, Klaus-Dirk (2014). Terminology and Localization. I: *Handbook of Terminology*, volym 1, 451–463. Red. Hendrik J. Kockaert & Frieda Steurs. John Benjamins Publishing Company.
- Spinuzzi, Clay & Mark Zachry. (2000) Genre ecologies: An Open-System Approach to Understanding and Constructing Documentation. I: *ACM Journal of Computer Documentation*. August 2000/Vol. 24, No 3, 169–181. [online] [citerat 27.3.2015]. Tillgänglig: <http://williamwolff.org/wp-content/uploads/2011/09/p169-spinuzzi.pdf>.
- Virtaluoto, Jenni (2013) “It’s a strange little business” – issues in technical communication. I: *AFinLA-e Soveltavan kielitieteen tutkimuksia* 2013/no. 5, 200–213. Red. Tiina Keisanen, Elise Kärkkäinen, Mirka Rauniomaa, Pauliina Siitonen & Maarit Sironmaa. [online]. [citerat 27.3.2015]. Tillgänglig: <http://ojs.tsv.fi/index.php/afinla/issue/view/1107>.