

1

Johdanto: Käytettävyyttä etsimässä

Vuonna 1991 IBM:n käytettävyysslaboratorio testasi Forte Travellodge – hotelliketjun puhelinvarauskeskuksen atk-järjestelmän. Tutkimuksessa löydettiin 62 käytön ongelmaa. Niiden korjaaminen lisäsi keskuksen kapasiteettia 80%:lla.

Lehdistö kertoi tällaisista uutisista mielellään! Lähes kaikilla heidän lukijoillaan on ollut turhauttavia kokemuksia vaikeasta tai jopa käsittämättömästä tekniikasta. Tarjoaisiko käytettävyyden testaus vihdoinkin lääkkeet tähän yleismaailmalliseen ongelmaan?

Käytettävyysslaboratorioissa pannaan tuoreita käyttäjiä suorittamaan annettuja tehtäviä. Heidän suorituksensa taltioidaan ja tutkitaan jälkikäteen perusteellisesti. Tuntuu järkeenkäyvältä, että sovelluksesta tulee helpompi käyttää, jos käyttöliittymää koskevat päätökset perustuvat havaintoihin eivätkä sovelluskehittäjien arvauksiin!

Vuosikymmenen loppupuolella julkisuudessa alkoi näkyä epäluuloja siitä, onko viisasten kiveä sittenkään löydetty. Monet ohjelmistotalot esittelivät mielellään julkisuudessa käytettävyysslaboratorioitaan, mutta käyttäjät alkoivat jupista, että missä ihmeessä se näkyy! Käsittämättömät so-

vellukset eivät ole hävinneet historiaan – ne aiheuttavat käyttäjäkunnalleen edelleen joka päivä stressiä ja kustannuksia.

Microsoft on mielellään esitellyt julkisuudessa Word-tekstinkäsittelyohjelman käytettävyydestä. Se on varmaankin alan historian testatuin sovellus. Mutta siitä huolimatta en muista koskaan lukeneeni tai kuulleeni jonkun kiittävän sen käytettävyyttä!

Kun Microsoft kävi oikeutta monopolisyytteistä, eräs Time-lehden lukija ehdotti yleisönosastossa, että sovelias rangaistus Bill Gatesille olisi panna hänet muotoilemaan jotain lakitekniistä asiakirjaa firmansa tekstin-käsittelyohjelmalla! Itse en vieläkään saa aina Wordia tekemään mitä haluan, ja joskus Word myös tekee omin päin asioita, joita en halua! Olen taittanut kokonaisia kirjoja painovalmiiksi WordPerfectillä, Ami Professionalilla ja Pagemakerilla. Mutta en haluaisi edes yrittää niin vaativaa tehtävää Wordilla, vaikka sen ominaisuudet periaatteessa kyllä riittäisivät siihen.

Tässä ei ole kyseessä mikään periaatteellinen Microsoft-vastaisuus. Mielestäni Microsoftin Powerpointin ja Excelin käytettävyyden on jokseenkin tyydyttävällä tasolla. Olen kuitenkin sitä mieltä, että Wordin käytettävyyden ei ole käytettävyydestä huolimatta kehittynyt parempaan päin.

Käytettävyyden testauksen ansiot ovat kuitenkin aivan kiistattomat! On aivan riittävästi Forte Travellodgen kaltaisia esimerkkejä siitä, että käytettävyyden testaus on tuonut numeroilla mitattavia parannuksia esimerkiksi toiminnan tehokkuuteen.

Näiden menestystarinoiden raportointi alan lehdistössä on kuitenkin saattanut levittää sellaista mielikuvaa, että käytettävyyden luodaan testamalla ja parantelemalla sovelluksen viimeistelyvaiheessa suunnittelutyön tuloksia.

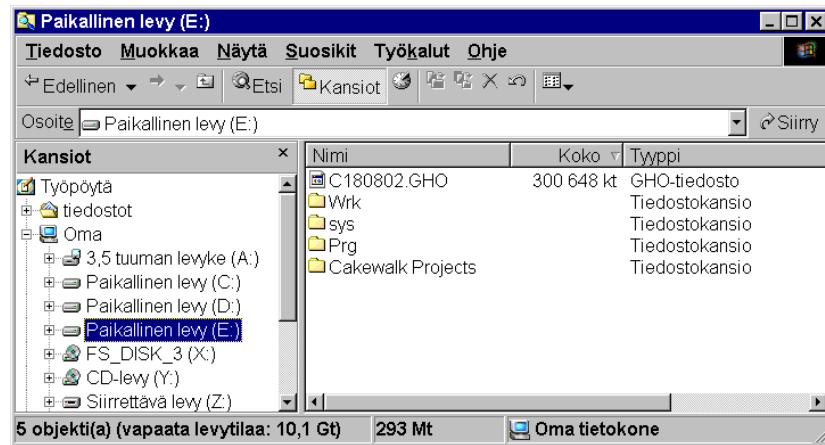
Testaus on todistettavasti käyttökelpoinen lääke käytettävyyden ongelmiin. Mutta hyvää käytettävyyttä ei luoda vain viimeistelyssä. Käytet-

tävyiden kysymysten täytyy olla mukana näkökulmana kaikissa sovelluksen suunnittelun vaiheissa!

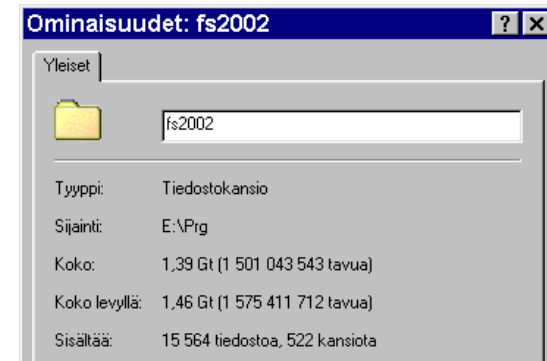
Erään käytettävyysongelman anatomia

Onko sinulta koskaan alkanut levytila loppua mikrostasi? Useimmilta mikronkäyttäjiltä on!

Windows tarjoaa tällaisiin levyn järjestelyihin Resource Manager – työkalun:



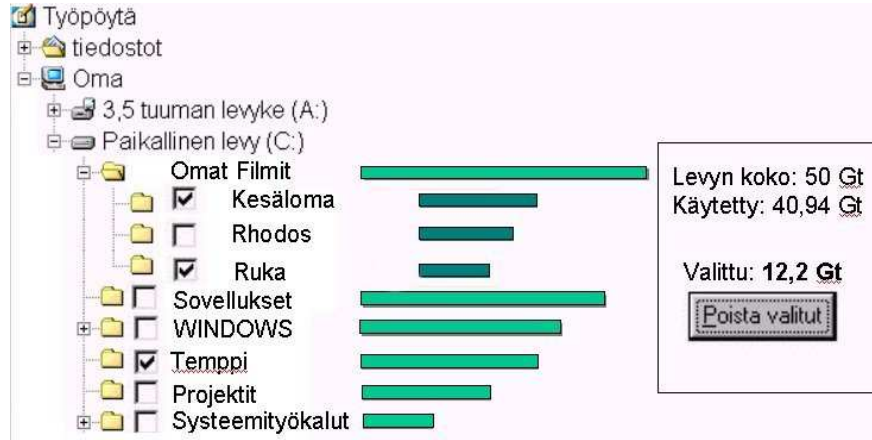
Kuvitellaan, että sinun pitäisi vapauttaa esimerkiksi kymmenen gigatavua tilaa. Yhden hakemiston poistaminen ei riitä; niitä täytyy poistaa useampia. Tästä Windowsin Resource Managerista on aika vähän apua tässä tilanteessa. Ikkunan alapalkista näet, paljonko levyllä on tilaa jäljellä. Sen sijaan et näe, miten paljon kukin hakemisto vie tilaa. Tämä tieto löytyy hakemistokohtaisista ominaisuusdialogeista:



Hakemiston koko on siis hankalasti piilossa! Kokenut käyttäjä osaa hyödyntää ominaisuusdialogeja, mutta aloittelevalla käyttäjällä Resurssienhallinnan perusikkunassa ei ole edes mitään vihjettä siitä, mistä hakemiston kokotietoa voisi etsiä.

Vaikka tämän tiedon löytäminen ei tuottaisi ongelmia, kymmenen gigan vapauttaminen olisi hyvin vaivalloista. Sinun täytyisi avata jokaisen hakemiston ominaisuusdialogi, sekä muistaa tai laittaa muistiin kunkin hakemiston koko. Sitten sinun pitäisi laskea päässä tai muulla tavalla niitä yhteen niin, että tarvittavat kymmenen gigaa saadaan kasaan. Lopulta sinun pitäisi viedä kaikki valitut hakemistot roskikseen.

Levytilaa vapauttava käyttäjä voisi olla kiitollinen, jos hänellä olisi esimerkiksi tämän tyyppinen dialogi käytettävissään:



Tämä esimerkki tarjoaa levytilan raivausta palvelevan näkökulman Explorerin sisältämään informaatioon. Hakemisto on järjestetty tasoittain suuruusjärjestykseen – suurimmat tilanviejät ovat ylimpänä. Palkki kertoo, kuinka suuren osan levyn tilasta kukin hakemisto vie. Käyttäjä voi ruksaamalla valita hakemistoja poistolistalle, ja oikealta painikkeen yläpuolelta hän näkee välittömästi, kuinka paljon on valittu poistettavaksi.

Microsoft on aivan varmasti testannut Windowsin käytettävyyttä ahkerasti. Mutta siitä huolimatta sen käytettävyys on kerta kaikkiaan huono tilanteessa, jossa käyttäjä haluaa vapauttaa tietyn määrän levytilaa. Missä siis vika?

Käytettävyyden testauksessa käyttäjä yrittää suorittaa annettuja tehtäviä. Testien laatijan täytyy siis ymmärtää käyttäjän toimintaa ja pyrkimyksiä!

Resource Managerin käyttö on melko vaivatonta ja havainnollista silloin, kun käyttäjä haluaa poistaa tietyt nimeltä mainitut hakemistot. Tällaista tilannetta on varmaankin hiottu Microsoftin käytettävyyslaboratoriossa. Mutta voisiko olla, että Microsoftilla on jäänyt miettimättä, kuinka vaivatonta on määrätyn suuruisen levytilan vapauttaminen? On varmaankin testaamattakin selvää, että parantamisen varaa on!

Ongelmat eivät tässä tapauksessa ilmeisesti liity käyttöliittymän viimeistelyyn vaan käyttäjän toiminnan ymmärtämiseen! Käyttöliittymän suunnittelijoilla ei ole ollut kaikkia oikeita kysymyksiä vastattavanaan.

Takaisin perusasioihin!

”Miten autan käyttäjää vapauttamaan halutun määrän levytilaa?” Tämä ei todellakaan ole vaikea kysymys! Yllä on esitetty yksi mahdollinen ratkaisu. Jokainen sovelluskehityksen ammattilainen osaisi varmasti nopeasti hahmotella muutamia muita käyttökelpoisia – ehkä parempiakin – vaihtoehtoja.

Oikeiden kysymysten löytäminen on usein paljon vaikeampaa kuin niihin vastaaminen! Niiden kysyminen perustuu siihen, että ymmärretään käyttäjien käsitteet, pyrkimykset, tilanteet ja tarpeet.

Näiden asioiden selvittäminen on perinteellisesti ollut keskeinen asia myös tietojärjestelmien suunnittelussa. Niinpä hyvin tehty systeemianalyysi ja -suunnittelu edesauttaa käyttäjätavallisten sovellusten syntymistä. Hyväkään käyttöliittymän suunnittelu ei sen sijaan pelasta sovelluksen käytettävyyttä, jos systeemisuunnittelu on tehty huonosti.

Termillä ”systeemisuunnittelu” ei tässä kirjassa tarkoiteta vain tietojärjestelmien suunnittelua. Sillä tarkoitetaan sitä työtä, joka tuottaa käyttöliittymän suunnittelussa tarvittavan ymmärryksen sovelluksen tarkoituksesta ja tehtävistä sekä käyttäjien käsitteistä, tilanteista ja tarpeista.

Kaikki sovellukset eivät ole tietojärjestelmiä, mutta tietojärjestelmien suunnittelun menetelmissä on paljon sellaisia käyttäjien tekemisten ja pyrkimysten analysoimisen työkaluja, jotka ovat sovellettavissa muunkin tyyppisiin sovelluksiin. Kaikilla sovelluksilla on käyttäjänsä, joilla on pyrkimyksiä, joihin liittyy käyttötilanteita ja tarpeita. Niiden selvittämisessä ei koskaan voi olla liian järjestelmällinen tai perusteellinen.

Monet sovelluskehityksen ja systeemityön mallit tekevät eron systeemin analyysin ja suunnittelun välillä. Analyysin tavoite on ymmärtää sovellusaluetta ja käyttäjän tarpeita. Systeemisuunnittelu taas tuottaa suunnitelman siitä, mitä tietoa uusi sovellus käsittelee, ja mitä se sille tiedolle

tekee. Käytännössä tällainen jako on kuitenkin vaikeaa, koska suunnittelun tulokset usein suuntaavat analyysiä.

Toisaalta UML-kirjallisuudessa analyysin ja suunnittelun raja sijoitetaan yleensä eri paikkaan: analyysi tuottaa uuden systeemin periaateratkaisun, ja suunnittelu kuvaa systeemin toteutuksen.

Tässä kirjassa ei ole tarpeen tehdä eroa analyysin ja systeemisuunnittelun välille. Nämä asiat niputetaan yhteen nimikkeen ”systeemisuunnittelu” alle. Systeemisuunnittelu pitää siis sisällään analyysin.

Jos systeemisuunnittelua ei ole tehty kunnolla, sovelluksen käytön suunnittelijat saattavat etsiä oikeita vastauksia väärin kysymyksiin, tai heiltä voi puuttua tärkeitä oikeita kysymyksiä. Jos näin on, käytettävyyden testaukseen ei perustu oikeisiin kysymyksiin.

Käytettävyyden kehittämisessä voidaan päästä hyvään tulokseen vain, jos suunnitteluprosessin perusasiat ovat kunnossa! Käytettävyydestä voi viimeistellä hyvän suunnittelun tulokset, mutta se ei juurikaan pysty korvaamaan suunnitteluprosessin puutteita.

Tässä kirjassa tullaan esittämään useampia esimerkkejä siitä, miten systeemisuunnittelun ongelmat aiheuttavat käytettävyyden ongelmia. Yritämme myös ymmärtää, että miten systeemisuunnittelun menetelmiä pitäisi käytännössä soveltaa, jotta käyttöliittymän suunnittelijoilla olisi mahdollisimman hyvät lähtökohdat.

Käytettävyys on käyttöliittymää isompi asia

PC-mikrojen arkkitehtuurin vahvuus on alusta asti ollut avoimuus. Käyttäjät ovat voineet asentaa mikroihiinsa eri valmistajien lisälaittekortteja, kuten näytönohjaukskortteja, paikallisverkkokortteja, modeemikortteja, äänikortteja, peliohjainkortteja jne.

Jotkut lukijat muistanevat vielä, millaista näiden korttien asentaminen oli ennen Windows 95:n aikaa. Käyttäjä joutui tekemään kaikenlaisia teknisiä asetuksia, kuten IRQ-asetuksia sekä DMA-asetuksia. Asetukset tehtiin usein piirikorteille ns. jumppereilla tai mikrokytkimillä.

Esim. keskeytyslinjoja (IRQ) oli käytettävissä kuusitoista kappaletta. Osa niistä oli varattu tietokoneen perustoiminnoille, mutta osa oli vapaana lisäkorttien käyttöön. Uutta korttia asentavan käyttäjän piti siis tietää, mitkä linjat yleensä ovat vapaana. Hänellä piti myös muistissa olla mahdollisesti aikaisemmin asennettujen korttien varaamat keskeytyslinjat, jottei hän yrittäisi antaa samaa linjaa kahdelle kortille.

Kaiken huipuksi näiden korttien asennusohjeet olivat yleensä aika käsittämättömiä muille kuin elektroniikkainsinööreille!

Ymmärrettävyys on keskeinen käytettävyyden tekijä. IRQ:t ja DMA:t olivat varmasti käsittämättömiä asioita käyttäjien enemmistölle. Tuon ajan PC:t olivat siis korttien asentamisen suhteen erittäin käyttäjäepäystävällisiä.

Vaivattomuus on toinen tärkeä käytettävyyden tekijä. Tällainen teknisten asetusten kanssa näpelöinti jos mikä on vaivalloista!

Hyvää käytettävyyttä eli käyttäjätavallisuutta etsitään usein käyttöliittymän yksityiskohdista. Mutta eräs IT-historian suurimmista käyttäjätavallisuuden edistysaskelista ei kuitenkaan liity käyttöliittymään: Windows 95:n Plug and Play -systeemi automatisoi lisäkortin asennuksen teknisten asetusten teon. Nyt käyttäjän tarvitsee vain laittaa kortti paikalleen sekä antaa koneelle kortin mukana tullut levyke. Windows kysyy kaikilta lisäkorteilta, että mitä keskeytyslinjoja ja muita teknisiä resursseja ne tarvitsevat, ja tekee sitten tarvittavat asetukset automaattisesti.

Käytettävyys on siis käyttöliittymää laajempi asia!

Plug and Play –esimerkki osoittaa myös, että hyvän käytettävyyden saavuttaminen vaatii usein laajoja systeemitason ratkaisuja. Korttien asennuksen automaatio toimii vain, jos kortit osaavat kertoa Windowsille, mitä ne tarvitsevat. Microsoftin on siis täytynyt vetää korttien valmistajat mukaansa tähän hankkeeseen.

Kohti laajempia näkökulmia

90-luvun alkupuolella käytettävyysalan kirjallisuus keskittyi paljolti

- hyvien käyttöliittymien periaatteisiin,

- ihmisen ja sovelluksen vuorovaikutuksen mallinnusmenetelmiin, sekä
- käytettävyyden testaamiseen.

Vuosituhanne lopulla alkoi näkyä merkkejä laajemmasta näkökulmasta käytettävyyden kysymyksiin. Esimerkiksi vuonna 1999 julkaistussa kirjassa ”The Usability Engineering Lifecycle” Deborah Mayhew katsoo sovelluskehityksen kaikkia osaprosesseja käytettävyyden näkökulmasta. Hän liittää hyvän käytettävyyden luomisen ketjuun myös liiketoimintaprosessien uudelleenrakentamisen (business process re-engineering) sekä työtehtävien uudelleenrakentamisen (work re-engineering). Usein tietotekniikka luo tilaisuuksia muuttaa kokonaisia prosesseja niin, että käyttäjän toiminta tulee paljon entistä yksinkertaisemmaksi.

Myös Teppo Tossavaisen väitöskirja ”The System Usability of Complex Technical Systems” vuodelta 2002 laajentaa käytettävyyden näkökulmaa. Hän toteaa, että ajattelu käytettävyydellä on tähän asti ollut hyvin käyttäjäkeskeistä. Hän perustelee kirjassaan näkökulman laajentamista käyttäjäkeskeisistä kokonaisten teknisten systeemien kehittämisen ja hyödyntämisen kaikkiin puoliin.

Mitä on tulossa

Tämä kirja on pyritty laatimaan niin, että kaikkien lukijoiden ei tarvitsisi lukea koko kirjaa. Liikkeelle lähdetään varsin laajasta näkökulmasta, jota sitten vähitellen ”zoomataan” kohti konkreettisia käytön suunnittelun kysymyksiä.

Luvuissa 2-4 katsomme, miten käytettävyyteen liittyvät asiat vaikuttavat yrityksen sekä sovellustuotannon ja projektien johtamiseen. Kaikkien lukijoiden kannattaa lukea luvusta 2 löytyvä käytettävyyden määritelmä. Muuten nämä luvut voi hypätä yli, jos oma kiinnostus on enemmän suunnittelun käytön suunnittelun konkreettisissa kysymyksissä.

Luvuissa 5-7 katsomme, miten systeemisuunnittelu voi jo ennen käyttäjäkeskeisten suunnittelua auttaa hyvän käytettävyyden saavuttamista. Eriytisesti luvut 5-6 ovat suositeltavaa luettavaa kaikille lukijoille.

Luvuissa 8-12 käymme läpi käyttäjäkeskeisten suunnittelun eväitä. Ensin katsomme käyttäjän ja sovelluksen välisen viestinnän peruseräitä. Sitten käsittelemme graafisen käyttäjäkeskeisen rakennusaineiden hyviä käyttötapoja. Käymme myös läpi kaikille sovelluskehittäjille tarpeellisia typografian peruseräitä. Nämä luvut voi jättää vähemmälle huomille, jos oma kiinnostus kohdistuu lähinnä sovelluskehityksen johtamisen kysymyksiin.

Luvussa 13 tarkastelemme käyttäjäkeskeisten suunnittelun prosessia sekä siihen elimellisesti liittyvää käytettävyydestausta. Se on tarpeellista luettavaa projektin vetäjille sekä käytön suunnittelusta vastaaville.

Yhteenveto

Käytettävyyden kirjallisuudessa on käsitelty paljon käyttäjäkeskeisten suunnittelua sekä käytettävyydestausta. On toki tärkeää, että esimerkiksi lomakkeet ovat selkeitä ja helposti luettavia, tai että valikot ovat ymmärrettäviä ja nopeita käyttää.

Tässä kirjassa etsitään kuitenkin käytettävyyden tekijöitä laajemmalla näkökulmalla. Jo ennen käyttäjäkeskeisten suunnittelua tapahtuu paljon selästä, mikä voi auttaa tai haitata hyvän käytettävyyden tavoittelua:

- Systeemanalyysin ja –suunnittelun täytyy eväitä käyttäjän suunnittelija ymmärtämään käyttäjän käsitteitä ja pyrkimyksiä.
- Systeemin toiminnallisuutta ja arkkitehtuuria koskevat päätökset sanelevat ne puitteet, joiden sisällä käyttäjäkeskeisyyttä tavoitellaan.