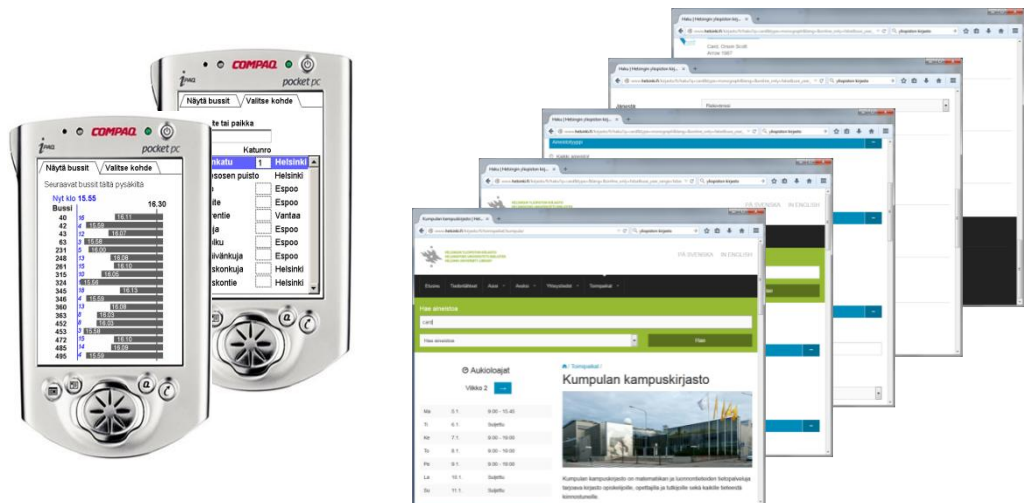


Simulointipohjainen käyttöliittymäsuunnittelu

Sari A. Laakso



Helsinki 13.1.2015

Kurssin luentomoniste

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1 Käyttöliittymän ominaisuudet	1
1.1 Käyttötilanteet ohjelmiston käyttöönottovaiheessa	1
1.2 Erityyppiset käyttöliittymäongelmat (Kuva-arkisto).....	2
Tehokkuus	3
Opittavuus.....	3
Sekundääriset ongelmat	4
Kulmakivet	5
Puuttuva toiminnallisuus tai tietosisältö	5
1.3 Määritelmiä kirjallisuudesta.....	6
1.4 Simulointimenetelmien idea	7
1.5 Esimerkki tehokkuusongelmasta (Asiakasrekisteri)	9
2 Käyttöliittymän arviointi.....	10
2.1 Simulointitestaus.....	10
Vaihe 1. Testitapausten selvittäminen.....	11
Vaihe 2. Vaihtoehtojen ja loppuratkaisun selvittäminen	12
Vaihe 3. Oikean polun selvittäminen	13
Vaihe 4. Kuvasarjan laatiminen	15
Vaihe 5. Testitulokset	18
Muita simulointipohjaisia asiantuntija-arviomenetelmiä.....	22
2.2 Käytettävyydestä	23
Testitehtävät.....	24
Testikäyttäjät	26
Muut testivalmistelut	27
Testitilanteen ohjaaminen	27
Testitulokset	30
Kustannustehokas testaus	33
3 Käyttötilanteet	34
3.1 Käyttötilanteet testitapauksina.....	34
3.2 GDD:n käyttötilanteet	36
3.3 Käyttötilanteiden kerääminen haastatteluilla	38
4 Käyttöliittymän suunnittelu	40
4.1 Erilaisia suunnitteluprosesseja	41
4.2 GUIDe-malli ja GDD-suunnitteluprosessi	44
4.3 Käyttöliittymän piirtäminen GDD:llä	47
Suunnitteluperiaatteet	48
Simulointi	49
Ongelmalistan käyttö	53
Esimerkki 1: PDA-bussiaikataulut	55
Esimerkki 2: Kirjastohakujärjestelmä.....	64
Lähteitä	74

Käyttöliittymän ominaisuudet

Viimeistään ohjelmiston käyttöönottovaiheessa järjestelmän avulla ryhdytään tekemään oikeita työtehtäviä ja suorittamaan todellisia käyttötilanteita. Tällöin puutteet ohjelmiston hyödyllisyydessä ja käytettävyydessä tulevat viimeistään esille.

Käyttötilanteet ohjelmiston käyttöönottovaiheessa

- Monesti ohjelmisto toteutetaan ja otetaan käyttöön ilman minkäänlaista käyttötilanteiden suorittamiseen (todellisen työn tekemiseen) perustuvaa arviointia tai testausta.
- Tällöin ohjelmiston käyttökelpoisuus tavallaan testataan vasta kentällä, ja ongelmien korjaaminen on kallista.



Sari A. Laakso (2015)

Käyttötilanteet ohjelmiston käyttöönottovaiheessa

Esimerkkejä ongelmista:

- Työtehtävien suorittaminen on vaivalloista ja aikaavievää.
- Kaikki työkulut eivät edes 'mene läpi', vaan osa vaiheista on suoritettava kiertoteitse paperilla tai muilla ohjelmistoilla.
 - Järjestelmästä puuttuu toimintoja.
 - Järjestelmästä puuttuu työtehtävissä tarvittavaa tietosisältöä, tai tiedot on organisoitu käyttäjän työtehtävän kannalta väärin.
- Käyttäjät eivät keksi, miten järjestelmää pitäisi käyttää. Käyttäjät tekevät turhaan virheitä, ja koulutukseen kuluu paljon resursseja.
- Järjestelmässä ylläpidetään toimintoja tai tietoja, joita kukaan käyttäjä ei oikeasti missään tilanteessa tarvitse.

Sari A. Laakso (2015)

Erityyppiset käliongelmat Esimerkki: Kuva-arkisto

- Kuvatoimittaja täydentää Paavo Lipposesta kertovaa lehtijuttua sopivilla kuvilla.

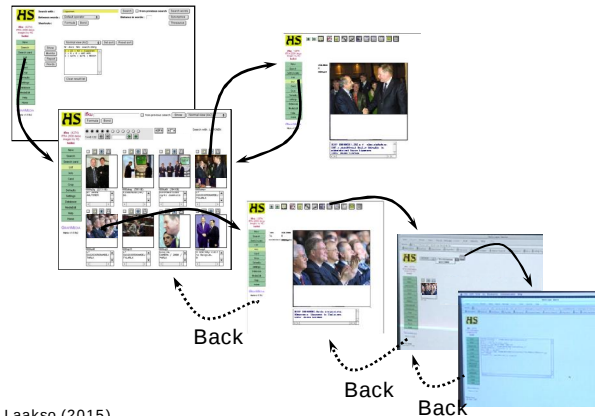


Sari A. Laakso (2015)

Tehokkuus:

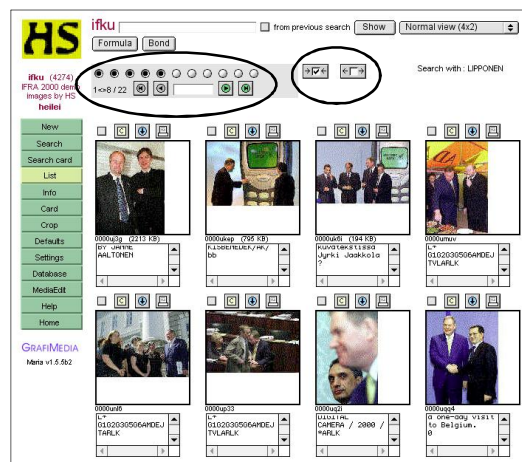
1. *Mekaaninen työ:* Käyttäjä joutuu tekemään valtavasti turhia edestakaisia navigointitoimenpiteitä.
2. *Kognitiivinen työ:* Käyttäjä joutuu pitämään mielessään tähän mennessä löytämiään hyviä vaihtoehtoja ja poissuljettuja vaihtoehtoja sekä näiden sijainteja.

Sopivien kuvien valitsemisen monivaiheinen polku:



Sari A. Laakso (2015)

Opittavuus: Käyttäjä ei ymmärrä ohjelman toimintalogiikkaa eikä keksi, mitä ohjelman toiminnot tarkoittavat ja miten niitä käytetään.



Sari A. Laakso (2015)

Erityyppiset käliongelmat

Sekundääriset ongelmat

- Käyttäjän tekemien virheiden korjaamisesta seuraava turha työ on eri asia kuin ensisijaisiin tehokkuusongelmiin sisältyvä turha työ (mekaaninen tai kognitiivinen). Virheen korjaamiseksi vaadittu turha työ on *seurausta toisesta ongelmasta*, kun taas varsinainen turha työ sisältyy sisäänrakennettuna vaatimuksena tehtävän suorittamiseen.
- Sekundääriset ongelmat johtuvat jostain toisesta ongelmasta:
 - Virheet johtuvat tyypillisesti tehokkuusongelmista (esim. turha kognitiivinen työ) tai opittavuus- tai muistettavuusongelmista.
 - Muistettavuusongelmien takana on opittavuusongelmia.
 - Käyttäjän tyytyväisyys syntyy osin kaikkien neljän osa-alueen pohjalta (tehokkuus, opittavuus, muistettavuus, virheet). *Lisäksi tyytyväisyyteen vaikuttavat mm. visuaalinen ulkoasu ja ylipäättään käyttöliittymän esteettisyys. Näihin emme perehdy tällä kurssilla, mutta ne on tärkeää tiedostaa ja ottaa huomioon muilla tavoin.*

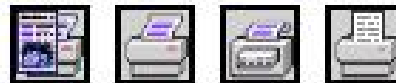
Sari A. Laakso (2015)

Opittavuus: Käyttäjän on vaikea päätellä neljästä samantapaisesta printteripainikkeesta, mistä tulostuisi hänen tarvitsemansa iso kuva.

Muistettavuus: Vaikka hän onnistuu löytämään oikean painikkeen ja ylittämään opittavuuskynnyksen, seuraavallakin kerralla on vaikea muistaa (palauttaa mieleen), mikä oli oikea toiminto.

Edellisten seurauksia: Em. opittavuus- ja muistettavuusongelmien takia käyttäjä helposti valitsee väärän toiminnon (virhe) ja joutuu palaamaan takaisin yrittämään uudelleen (turhaa työtä).

=>



Sari A. Laakso (2015)

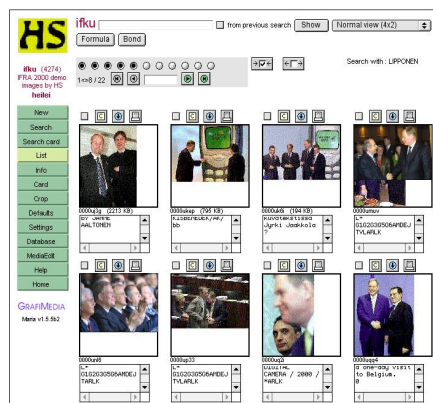
Erityyppiset käliongelmat Kulmakivet

- Nielsenin esittämät neljä ensimmäistä käytettävyyden osa-alueetta (tehokkuus, opittavuus, muistettavuus, virheet) näyttävät palautuvan kahteen kulmakiveen:
 - Tehokkuus: mekaaninen ja kognitiivinen työ
 - Opittavuus
- Muissa osa-alueissa näkyvät ilmiöt (virheet, muistettavuus, osin miellyttävyys) ovat valtaosin seurausta näistä.
- Opittavuus- ja muistettavuusongelmien sekä virheiden määrä saadaan minimoitua optimoimalla tehokkuus.
 - Tehokas käyttöliittymä sisältää vain vähän opittavuuskynnyksiä (työvaiheita on vähän ja käyttäjä on koko ajan yhdessä paikassa).
 - Käyttäjän on helppo korjata myös inhimillisten erehdysten seuraukset, koska kaikki näkyy koko ajan ja käyttäjä voi säätää ei-toivotut seuraukset samantien takaisin.

Sari A. Laakso (2015)

Puuttuva toiminnallisuus tai tietosisältö: Ohjelmasta puuttuu toimintoja tai tietoja, joita käyttäjä tarvitsi työkulkunsa osaksi.

Yksittäisten kuvien poimimistoiminto (ns. ostoskori) puuttuu:



= >

Seurauksia:

Käyttäjän on keksittävä kiertoteitä tehtävänsä hoitamiseksi, esim. tulostettava valitsemiaan kuvia paperille ja haettava myöhemmin samoja kuvia kuvanumeron avulla (turha työ). Työtä tulee valtavasti ja riski virheille kasvaa.

Sari A. Laakso (2015)

Määritelmiä kirjallisuudesta Käytettävyyden osa-alueet (Nielsen)

Käyttökelpoisuus (usefulness)

1. Hyödyllisyys (utility): Voiko tehdä oikeaa asiaa?

Onko järjestelmässä sellaiset **toiminnot** ja **tietosisältö** (Nielsen: vain toiminnot), että työtehtävien tekeminen 'menee läpi' – vaikka vaikeasti ja vaivalloisestikin?

Jos oikean asian pystyy tekemään jotenkin:

2. Käytettävyys (usability): Onko tekeminen sujuvaa?

Tehokkuus
(efficiency)

Onko turhia vaiheita, onko turhaa kognitiivista työtä

Opittavuus
(learnability)

Keksiikö käyttäjä, mitä pitäisi tehdä ja mitä tiedot tarkoittavat

Muistettavuus
(memorability)

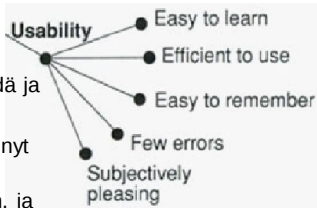
Onko selvää, kun on kerran keksinyt

Virhealttius
(errors)

Houkuttaako käli virhetoimintoihin, ja miten käyttäjä selviytyy virheistä

Tyytyväisyys
(satisfaction)

Kokeeko käyttäjä käytön miellyttävänä



[Nielsen93, s. 25; Nielsen03]

Sari A. Laakso (2015)

Ero Nielsenin määritelmään: Tehokkuus sidottu työn määrään

- Huomaa, että olemme aiemmin määritelleet tehokkuuden käyttäjältä vaadittavan mekaanisen ja kognitiivisen työn määrän avulla (vrt. *usage patterns* [Hornbæk 2006, s. 85]).
- Nielsen (ks. esim. [Nielsen 1993, s. 30-31]) sitoo tehokkuuden tehtävän suorittamiseen kuluvaan aikaan, esim. kuinka monta minuuttia testikäyttäjältä kuluu käyttötilanteen suorittamiseen (vrt. *task completion time* [Hornbæk 2006, s. 84]).
- Ensimmäinen perustuu käyttötilanteen suorittamisessa vaadittujen resurssien analysoimiseen ilman testikäyttäjiä: tietynlainen käyttöliittymäratkaisu vaatii käyttäjältä tietyt toimenpiteet ja 'miettimistyön', olipa käyttäjä millainen tahansa. Jälkimmäinen perustuu testikäyttäjien avulla saatavaan mitattavaan aineistoon: paljonko aikaa käyttäjiltä oikeasti meni.

Sari A. Laakso (2015)

Määritelmiä kirjallisuudesta ISO-standardi

Vastineet
vapaasti
käännetty;
termit on
yhdistetty
tällä kurssilla
käytettyihin
käsitteisiin.

- ISO 9241-11 –standardissa käytettävyyks on määritelty laajemmin:
Usability = Extent to which a product can be used by specified users to achieve specified goals (vrt. käyttötilanteet) with effectiveness, efficiency and satisfaction in a specified context of use.
 - Effectiveness: the accuracy and completeness with which specified users can achieve specified goals in particular environments
Tuloksellisuus: kuinka täsmällisesti ja kattavasti käyttäjä pystyy tekemään työnsä; onnistuuko eteen tulevien käyttötilanteiden suorittaminen kokonaan ja loppuun asti (vaikka olisi työlästä)
 - Efficiency: the resources expended in relation to the accuracy and completeness of goals achieved
Tehokkuus (kustannustehokkuus): kuinka paljon mekaanista ja kognitiivista työtä käyttäjän on tehtävä tilanteen suorittamiseksi
 - Satisfaction: the comfort and acceptability of the work system to its users and other people affected by its use
Tyytyväisyys: kuinka miellyttävä ja mieluinen järjestelmä on käyttäjille ja niille ihmisille, joihin sen käyttäminen vaikuttaa

Sari A. Laakso (2015)

Simulointimenetelmien idea Kulmakivistä suunnittelumenetelmä

- Tällä kurssilla opetettava simulointitestaus ja simulointipohjainen suunnittelumenetelmä GDD perustuvat tuloksellisuuden ja tehokkuuden (effectiveness, efficiency [ISO9241-11]) maksimointiin käyttötilanteita vasten (vrt. "specified goals", [ISO9241-11]). Opittavuus huomioidaan vasta lopuksi. Tämä riittää tuottamaan hyvän käyttöliittymän.
- Käyttötilanteiden suorittamisessa tarvittavat toiminnot ja tietosisältö (toiminnalliset vaatimukset ja tietosisältövaatimukset) tulevat simuloinnin kuluessa esille "itsestään", kun suunnittelija/arvioija hakee tilanteelle parasta loppuratkaisua ja pyrkii keskittymään tehokkuuden optimointiin. Hän valitsee sellaisia toimintoja ja tietosisältöä, jotka tukevat käyttötilannetta mahdollisimman suoraviivaisesti.
- Tehokkuudella ja opittavuudella operoidaan siten, että...
 - ensiksi optimoidaan tehokkuus (mekaaninen ja kognitiivinen) ja
 - vasta lopuksi korjailaan opittavuutta, jos tarpeen.

Sari A. Laakso (2015)

Simulointimenetelmien idea Kohti objektiivisuutta

- Riittävän optimaalinen käyttöliittymäratkaisu on mahdollista löytää käyttötilanteita vasten varsin objektiivisesti. Kun käyttötilanteet on valittu, niitä pystytään tukemaan paremmin tai huonommin erilaisilla käyttöliittymäratkaisuilla.
- Käyttöliittymäratkaisun arviointi muuttuu epämääräiseksi mielipidekysymykseksi heti, jos käyttötilanteet otetaan pois ja käyttöliittymiä tai sen osia yritetään tarkastella ilman käyttötilanteiden simulointia.
- Simuloitaessa erilaisten mielipiteiden esitleminen alkaa muuttua tarpeettomaksi ja korvautuu yhteisellä ongelmanratkaisulla, kun simulointia seuraavat henkilöt pyrkivät optimoimaan käsillä olevan tehtävän suorittamista käyttöliittymän avulla.

Sari A. Laakso (2015)

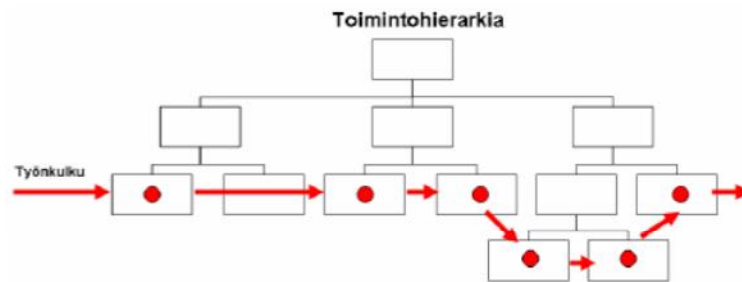
Simulointimenetelmien idea Toimintojen ja tietosisällön rooli

- Käytettävyyttä ei voida lisätä virheellisesti määritellyn tietosisällön tai toiminnallisuuden päälle.
- Tietosisällön ja toiminnallisuuden määrittely on osa käyttöliittymäsuunnittelua, ja se sisältyy tällä kurssilla käsiteltäviin simulointimenetelmiin.

Sari A. Laakso (2015)

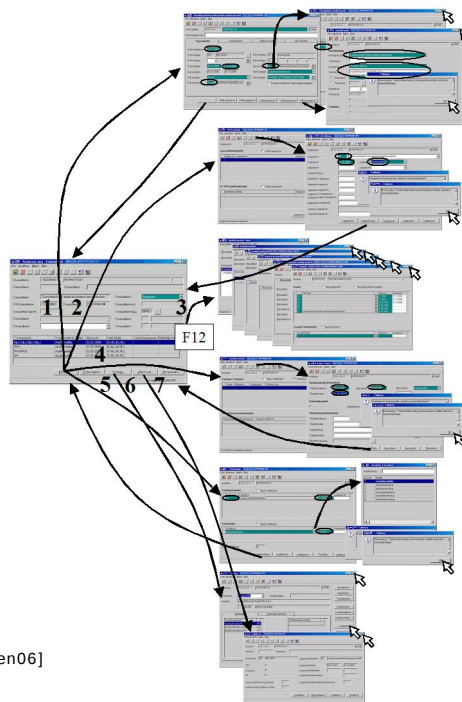
Esimerkki tehokkuusongelmasta: Asiakasrekisteri

- Tämän esimerkkijärjestelmän (asiakasrekisteröinnin sovellus) kaikki toiminnot suunniteltiin ilman käyttötilanteita. Osa toiminnoista saatiin vanhasta merkkipohjaisesta järjestelmästä.
- Aluksi laadittiin hierarkkisesti organisoitu toimintoluettelo, jonka pohjalta käyttöliittymä suunniteltiin. Eri hierarkiahaarojen toiminnot päätyivät tyypillisesti omiin ikkunoihinsa.
- Tämän tuloksena yhden käyttötilanteen suoritus (työnkulku) kulkee monen eri ikkunan kautta. Suorituspolku on pitkä ja monimutkainen => sekä käyttöliittymän tehokkuus että opittavuus ovat huonoja.



Asiakasrekisteri (jatkuu)

- Tässä tarkastellaan yhden tyypillisen, yksinkertaisen käyttötilanteen suorittamista em. sovelluksella.
- Kuvan tilanteessa käyttäjä lisää uuden asiakkaan asiakasrekisteriin.
- Kuvaan on visualisoitu tehtävän suorittamisessa tarvittavat näytöt ja näyttöjen välinen navigointitarve.



Käyttöliittymän arviointi

Testauksen tarkoituksena on löytää mahdollisimman paljon käyttöliittymän ongelmakohtia tai esimerkiksi asettaa erilaisia käyttöliittymäratkaisuja paremmuusjärjestykseen.

Käytettävyystestaus on hyvin tunnettu, kirjallisuudessa kuvattu testausmenetelmä, jossa testikäyttäjät käyttävät arvioitavaa järjestelmää. Simulointitestaus on kehitteillä oleva menetelmä, jonka suorittaa asiantuntija, ilman testikäyttäjiä.

Molemmissa menetelmissä testitapauksena on realistisia käyttötilanteita (työtehtäviä), joita lähdetään suorittamaan.

Simulointitestaus

Simulointitestaus on kehitteillä oleva käyttöliittymän arviointimenetelmä, joka on käytössä joissain yrityksissä (lisätietoja: Sari A. Laakso, salaakso@cs.helsinki.fi).

Menetelmässä on yhteisiä piirteitä muiden, käyttötilanteiden suorittamiseen perustuvien menetelmien kanssa.

Menetelmä löytää tehokkuusongelmien lisäksi puuttuvaa tietosisältöä ja toiminnallisuutta. Kantavana ideana on paikantaa päätöksenteossa tarvittava tieto ja minimoida turha työ (mekaaninen ja kognitiivinen työ).

Simulointitestauksen tekeminen

Yleiskuva menetelmästä

- Lähtökohdaksi arvioija selvittää jonkin tosielämässä käyttäjälle eteen tulevan konkreettisen käyttötilanteen.
- Arvioija selvittää, mikä olisi valitussa käyttötilanteessa käyttäjälle paras loppuratkaisu riippumatta siitä, millä järjestelmillä tilanne hoidetaan.
- Seuraavaksi arvioija selvittää, mikä on käyttöliittymän tarjoama oikea polku parhaan loppuratkaisun saavuttamiseksi:
 - Mitkä mekaaniset toimenpiteet käyttäjän pitää tehdä
 - Mitkä kognitiiviset toimenpiteet käyttäjän pitää tehdä:
 - mitkä näytöltä poimittavat tiedon palaset vaikuttavat hänen päätöksentekoonsa,
 - mitä tiedonkäsittelyä hänen on tehtävä päässään, esim. laskutoimituksia, vaihtoehtojen suhteuttamista toisiinsa ja päätösten tekemistä.
- Arvioija laatii oikeasta suorituspoluta kuvasarjan.
- Arvioija poimii kuvasarjasta käyttöliittymän parannusehdotuksen tai raportoi ongelmakohdat.

Sari A. Laakso (2015)

Vaihe 1:

Testitapausten selvittäminen

- Lähtökohdaksi arvioija selvittää ne konkreettiset käyttötilanteet, joita käyttäjän pitäisi pystyä järjestelmällä suorittamaan. Tähän käytetään käyttäjien työn havainnointia, kontekstuaalisia haastatteluja ja erillisiä käyttäjähaastatteluja.
- Käyttötilanteet jäsennetään niin, että tilanteen lähtöasetelma, aktivoitumishetki, ongelman virittävä ristiriita ja erityisesti käyttäjän tietämys ja puuttuva tietämys tulevat selvästi esille.
- Simulointitestaukseen arvioija valitsee tarkasteltavaksi yhden käyttötilanteen kerrallaan.

Sari A. Laakso (2015)

Vaihe 2: Vaihtoehtojen ja loppuratkaisun selvittäminen

- Arvioija selvittää, mitkä ovat käyttötilanteen päätöksentekovaiheessa relevantit vaihtoehdot, joiden välillä käyttäjän päätöksenteko tapahtuu. Esimerkiksi baletin väliaikatarjoilutilanteessa arvioidaan, mitkä leivokset voisivat tulla kyseeseen valitun käyttäjän tilanteessa, tai Pariisin matkaoppaan valintatilanteessa tutkitaan, mitkä kirjat sopisivat tähän tarkoitukseen parhaiten. Arvioija valitsee vähintään kolme hyvää vaihtoehtoa. Ei ole merkitystä, minkälaisen käyttäjän mieltymyksiin arvioija asettuu, mutta on tärkeää, että kyseessä on realistinen ja riittävän tyypillinen asetelma.
- Arvioija selvittää (käyttöliittymästä riippumatta) käyttötilanteeseen mahdollisimman hyvän loppuratkaisun: mihin relevanteista vaihtoehdoista käyttäjän juuri tässä tilanteessa kannattaisi päätyä, jos hän tietäisi kaiken sen, mikä tilanteessa on keskeistä tietää.
- Jos tilanteen suorittamiseen ei sisälly lainkaan päätöksentekoa, arvioija selvittää pelkän loppuratkaisun. (Yleensä tällöin käyttötilanteen suorittaminen voidaan automatisoida eli käyttöliittymä voidaan tältä osin poistaa.)

Sari A. Laakso (2015)

Vaihe 2 (jatkuu) Loppuratkaisun selvittämisestä

- Arvioijan on hankittava mahdollisimman kattavasti kaikki todelliseen tarjontaan liittyvä tieto, jotta hän pystyy päättämään, mikä olisi käsillä olevassa tilanteessa käyttäjälle paras (mahdollisimman hyvä) loppuratkaisu.
- Arvioijan tietämys on tässä eri asia kuin todellisen loppukäyttäjän tietämys realistisessa käyttötilanteessa.
- Järjestelmän avulla selville saatava paras ratkaisu voi olla eri kuin todellisuudessa paras loppuratkaisu. Menetelmässä haetaan todellisuudessa parasta (mahdollisimman hyvää) loppuratkaisua.
- Jos järjestelmän avulla selville saatava paras ratkaisu on huonompi kuin muilla keinoilla löytyvä ratkaisu, se saattaa johtua esimerkiksi käyttöliittymän tietosisältöongelmasta: järjestelmässä ei ole sellaista tietosisältöä, jonka avulla käyttäjä voisi löytää tilanteeseensa paremman ratkaisun.

Sari A. Laakso (2015)

Vaihe 3: Oikean polun selvittäminen

- Arvioija selvittää käyttötilannetta simuloimalla, mikä on käyttöliittymän tarjoama mahdollisimman yksinkertainen toimenpidepolku relevanttien vaihtoehtojen vertailemiseksi ja tilanteen suorittamiseksi loppuun:
 - Mitkä toimenpiteet käyttäjältä vaaditaan (mekaaninen työ)? Arvioija merkitsee tarvittavat toimenpiteet näyttökuviin.
 - Missä kohdissa käyttäjän on poimittava tietoja näytöltä, vertailtava vaihtoehtoja mielessään ja tehtävä päätöksiä (kognitiivinen työ)? Arvioija merkitsee näyttökuviin, mitä tiedon palasia käyttäjä hyödyntää päätöksenteossaan:
 - Mitkä tiedon palaset vaikuttivat päätöstä tehdessä siihen, että käyttäjä poimi tämän vaihtoehdon relevanttien joukkoon.
 - Mitkä tiedon palaset vaikuttivat siihen, että käyttäjä päätti sivuuttaa tämän vaihtoehdon.
 - Kuinka käyttäjä hakee puuttuvan tietosisällön toisesta järjestelmästä.

HUOM. Arvioija ei itse toimi tässä testikäyttäjänä vaan asiantuntijana.

Sari A. Laakso (2015)

Vaihe 3 (jatkuu) Menetelmän olettava tietämys

- Oikean polun suorittamisessa simuloidaan sellaista käyttäjää, jolla on
 - käyttöliittymän toimintalogiikasta epärealistisen täydellinen tietämys (ei simuloida todellisen käyttäjän mahdollista realistista harhailua ja väärrien toimintojen käynnistämistä),
 - mutta käyttöliittymän tietosisällöstä realistinen tietämys (esimerkiksi käyttäjä ei voi etukäteen tietää, onko ravintolassa tilaa haluttuna iltana, tai käyttäjä tietää, että McDonald's tarjoaa hampurilaisia ja pikaruokaa).
- Tarkoituksena on selvittää ensisijaisesti hyödyllisyys- ja tehokkuusongelmia, ei opittavuusongelmia.
 - Vrt. esim. kognitiivinen läpikäynti, jossa oikeaa suorituspolkua tarkastellaan käyttäjän ensimmäisen käyttökerran näkökulmasta eli käyttäjä ei tunne käyttöliittymän toimintalogiikkaa lainkaan → Opittavuusongelmia.
 - Menetelmät olettavat eri ääripäät toimintalogiikan tuntemuksesta.

Sari A. Laakso (2015)

Vaihe 3 (jatkuu)

Arvioijan vs. käyttäjän tietämys

Arvioijan oma tietämys, joka hänen on hankittava arviointia varten	Oletetun käyttäjän tietämys simulointisekvenssin alussa
Täydellinen tietämys käyttöliittymän toimintalogiikasta.	Täydellinen tietämys käyttöliittymän toimintalogiikasta.
Tuntee tilanteessa tarvittavan tietosisällön.	Ei tunne (tai tuntee vain osittain) tarvittavan tietosisällön ennestään.
Tietää tilanteen päätöksentekokohtien relevantit vaihtoehdot.	Ei tiedä (ainakaan kaikkia) relevantteja vaihtoehtoja ennestään.
Tietää tilanteeseen hyvän loppuratkaisun.	Ei tiedä tilanteen loppuratkaisua ennestään.

Arvioija selvittää toimenpiteet, joilla käyttäjä saa nämä selville.

Sari A. Laakso (2015)

Vaihe 3 (jatkuu)

Päätöksentekokohtien vertailu

- Kun testataan käyttäjän päätöksentekoa sisältäviä vertailutilanteita, arvioijan tulee simuloida riittävän kehittyneellä päätöksentekoprosessilla, jotta kognitiiviset ongelmat (mm. mielessä pitämisen ongelmat) saadaan esiin:
 - Testisekvenssissä käyttäjän tulee osua vähintään kolmeen relevanttiin vaihtoehtoon: käyttäjä punnitsee näitä keskenään ja arvioi, mikä olisi paras tässä tilanteessa.
 - Käyttäjälle tulee eteen myös useita huonoja vaihtoehtoja, jotka hän jättää sivuun.
- Jos arvioija testaa epärealistisen yksinkertaista päätöksentekoa (esim. käyttäjä menee Turkuun aamun ensimmäisellä junalla, oli se mikä tahansa) tai valitsee pelkästään helppoja erikoistapauksia (juna menee Kemijärvelle vain yksi päivässä), keskeiset ongelmakohdat eivät nouse esille testauksessa.

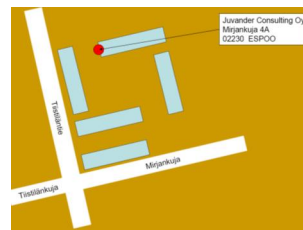
Sari A. Laakso (2015)

Vaihe 3 (jatkuu) Simulointi loppuun asti

- Simulointia ei pidä lopettaa vielä tietokoneohjelman käytön päättymiseen (esim. *kun käyttäjä on saanut varattua hotellihuoneen ja sulkee ohjelman*), vaan...
- ...se tulee viedä loppuun aina tavoitteen saavuttamiseen asti (esim. *kunnes käyttäjä on ajanut autolla perheineen hotelli Mesikämmeneen ja mennyt huoneeseensa*).
- Käyttötilanteen loppupään simulointi voi paljastaa yllättäviä ongelmakohtia käyttöliittymästä. Esimerkiksi ajokartan avulla ei oikeasti löydäkään perille, tai hotellihuone ei vastaakaan sitä, mitä varausjärjestelmä käyttäjälle lupasi.



Lähde: www.scandic-hotels.fi



Lähde: www.juvander.fi/yhteystiedot/Company_Location.pdf

Sari A. Laakso, Antti Latva-Koivisto (2006)

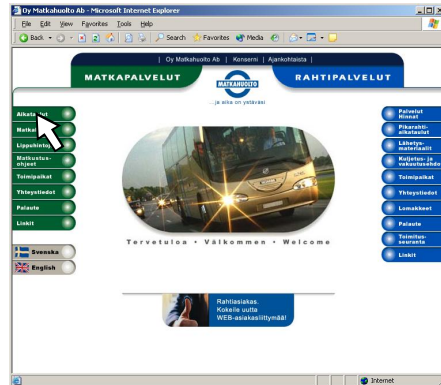
Vaihe 4: Kuvasarjan laatiminen

- Simulointitestauksen sisältämistä vaiheista kootaan kuvasarja, josta näkyy käyttäjän jokainen toimenpide. Kuvasarja auttaa arvioimaan käyttöliittymän ongelmakohtia, ja sen avulla on havainnollista esittää simuloinnin tuloksia.
- Kuvasarjan ensimmäisenä kuvana on järjestelmän alkutila, jossa käyttäjä ei ole vielä tehnyt ensimmäistäkään toimenpidettä (esim. etusivu www.matkahuolto.fi).
- Jokaiseen kuvaan merkitään yksi käyttäjän toimenpide, joka on tyypillisesti hiirellä klikkaaminen (merk. nuolikursorin kuvalla) tai tiedon syöttäminen näppäimistöltä (ympäröi syötteet). Lisäksi merkitään päätöksentekoon vaikuttavat tiedon palaset, jotka käyttäjä lukee näytöltä.
- Toimenpiteen seuraus eli järjestelmän reaktio esitetään seuraavassa kuvassa. Jos käyttäjä esim. täyttää peräkkäin monta syötekenttää, joihin ohjelma ei reagoi mitenkään, nämä toimenpiteet voidaan merkitä samaan kuvaan. Aina, kun ohjelma reagoi käyttäjän toimenpiteeseen jotenkin, otetaan uusi näytökuva.

Sari A. Laakso (2015)

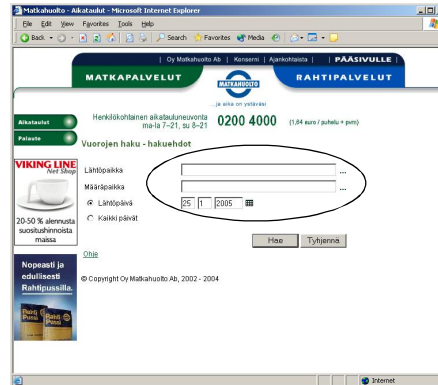
Esimerkkikatkelma kuvasarjasta: www.matkahuolto.fi

Huom. Alareunan tekstit ovat ohjeita, jotka eivät tule mukaan itse kuvasarjaan.



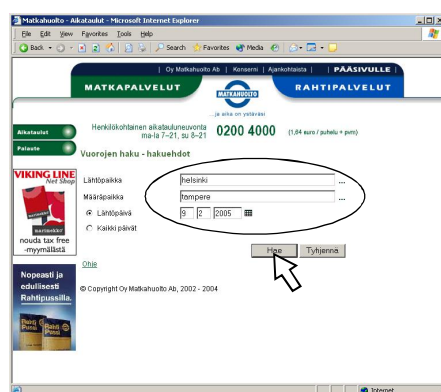
Alkutilakuvana on sellainen näytökuva, jossa käyttäjä ei ole vielä tehnyt ensimmäistään toimenpidettä. (Alkutilakuvaan voit merkitä ensimmäisen hiiren painalluksen nuolikursorin kuvalla.)

Sari A. Laakso (2006)



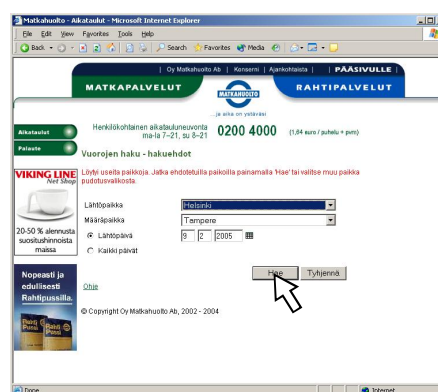
Näyttö avautuu tämän näköisenä, kun käyttäjä on tehnyt edellisen kuvan toimenpiteen eli painanut hiirellä *Aikataulut*-linkkiä vasemmasta reunasta. Tähän kuvaan merkitään ympäröinnillä kentät, joihin käyttäjä seuraavaksi syöttää tietoja.

Esimerkkikatkelma kuvasarjasta (sivu 2/3)



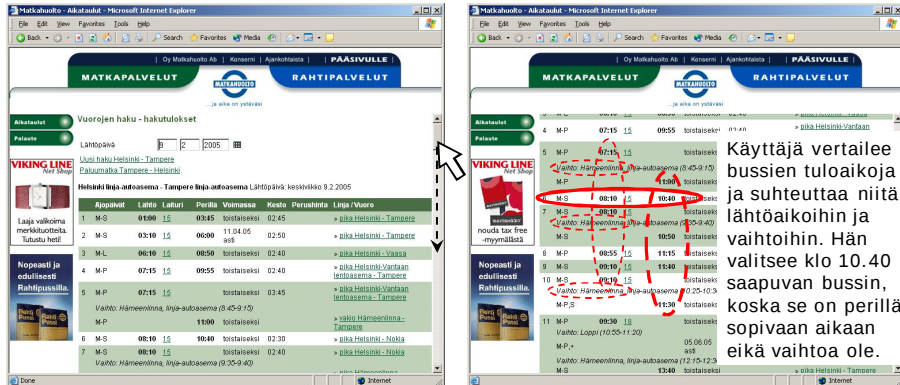
Tässä käyttäjä on syöttänyt ympäröityihin kenttiin *"helsinki"* ja *"tampere"* ja vaihtanut päivämäärän. (Tähän kuvaan voit merkitä myös *Hae*-painikkeen, koska ohjelma ei reagoi ennen sitä. Yhtä hyvin voit merkitä *Hae*-painikkeen vasta seuraavaan kuvaan.)

Sari A. Laakso (2006)



Kun käyttäjä on edellisessä kuvassa painanut *Hae*-painiketta, näytölle ilmestyy yllä näkyvä punainen teksti ja pari pudotusvalikkoa. Seuraavaksi käyttäjä painaa uudelleen *Hae*-painiketta.

Esimerkkikatkelma kuvasarjasta (sivu 3/3)



Seuraavaksi käyttäjä vierittää sivua alaspäin nähdäkseen ennen puoltapäivää perillä olevat bussit. Merkitse raahaus nuolikursorilla ja katkoviivalla, joka kuvaa raahaamisen suuntaa ja pituutta.

Tässä käyttäjä vertailee eri busseja ja yrittää päättää, millä bussilla hänen kannattaisi mennä. Merkitse tärkeimmät päätöksentekoon vaikuttavat tiedot kuvaan punaisella katkoviivalla. Kirjoita viereen tekstillä, mitä käyttäjä vertailee, minkä hän valitsee ja miksi. Merkitse valittu bussi punaisella ympyröinnillä.

Sari A. Laakso (2006)

Vaihe 4 (jatkuu) Vertailumerkinnöillä valintapäätökset

- Vertailumerkintöjen tekeminen auttaa löytämään sellaisia käyttöliittymän ongelmakohtia, jotka liittyvät käyttäjän mielessä tapahtuvaan vaihtoehtojen vertailemiseen ja päätöksentekoon.
- Vertailumerkinnöillä ympäröidään päätöksenteossa vaikuttavat yksittäiset tiedon palaset eli ne kohteet (sanat, kuvat, taulukon solut jne.), joiden perusteella käyttäjä päättää, että kyseinen vaihtoehto on hyvä tai huono. Älä ympäröi kaikkea dataa, jota käyttäjä voisi näytöltä lukea tai katsoa, vaan ainoastaan ne tiedon palaset, jotka juuri tässä käyttötilanteessa *vaikuttavat* valintapäätökseen.
- Puuttuvan tietosisällön voit simuloida haettavaksi toisesta järjestelmästä. Käytä ensisijaisesti arvioitavaa järjestelmää, mutta jos jokin tilanteen suorittamisessa tarvittava tieto puuttuu sieltä kokonaan, sisällytä kuvasarjaan toisen järjestelmän käyttö, josta käyttäjä poimii tarvittavan tiedon, esimerkiksi kännykän kalenteri tai Google Maps.

Sari A. Laakso (2015)

Vaihe 4 (jatkuu) Käyttäjän toimenpiteet

Merkitse käyttäjältä vaadittavat toimenpiteet näyttökuviin seuraavasti:



Hiirellä klikkaaminen (click). Merkitse nuolen kärki klikattavan kohteen päälle.



Tuplaklikkaaminen (double click).



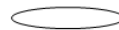
Hiiren oikealla napilla klikkaaminen (right click).



Hiiren kohdistaminen (mouseover) ja tooltip-infon ilmestyminen.



Hiirellä raahaaminen (drag). Raahauksen suunta ja pituus merkitään katkoviivalla.



Tiedon syöttäminen näppäimistöltä (mustalla viivalla ympäröinti). Käyttäjä syöttää tekstiä näppäimistöltä kenttiin, jotka on ympäröity.



Tiedon poimiminen katsomalla (punainen katkoviiva). Käyttäjä poimii tarvitsemiaan tiedonpalasia (tekstiä, kuvia) näytöltä mieleensä pelkästään lukemalla ja katsomalla. Tarvittavat tiedonpalaset on ympäröity punaisella katkoviivalla.

Sari A. Laakso (2015)

Vaihe 5: Testitulokset

- Testitulokset voidaan poimia vertailumerkinnöillä varustetusta kuvasarjasta. Menetelmän esille tuomista tuloksista voidaan poimia käyttötarkoituksen mukaan...
 1. tilanteen virittämä optimiratkaisu (ns. parannusehdotus) tai
 2. käyttöliittymän ongelmakohdat: puuttuvaan tietosisältöön tai toiminnallisuuteen liittyvät ongelmat sekä tehokkuusongelmat.

Sari A. Laakso (2015)

5.1 Tuloksena optimiratkaisu Suoraan parannusehdotukseen

- Optimaalisen käyttöliittymäratkaisun näet poimimalla relevanttien vaihtoehtojen välisessä vertailussa tarvitut tiedon palaset, jotka arvioija kerää sekä arvioitavasta järjestelmästä että muista ulkoisista järjestelmistä (puuttuva tietosisältö).
 - Tietosisältö: Ensiksi kaikki simulointisekvenssissä tarvitut *tiedon palaset* organisoidaan samalle näytölle kerralla näkyviin.
 - Toiminnot: Seuraavaksi muodostetaan yksinkertainen mahdollinen toimenpidesekvenssi, jolla käyttäjä voi säättää em. tiedot esiin järjestelmän alkutilasta käsin, samalla näytöllä. (Toimenpidesekvenssin logiikan on oltava aukoton. Järjestelmä ei voi maagisesti arvata, että käyttäjälle pitäisi näyttää juuri tämä data nyt.)
- Optimiratkaisu muodostuu periaatteessa samalla tavalla kuin GDD-suunnittelumenetelmässä, ks. päätöksentekolähtöinen simulointi: ks. s. 49 alempi kalvo ja selitykset s. 50-51). Ratkaisu virittyy käyttötilanteesta 'itsestään'.

Sari A. Laakso (2015)

5.2 Tuloksena käliongelmiä Ensin läpimeno, sitten tehokkuus

- Ensin katsotaan, meneekö käyttötilanteen suorittaminen läpi, eli onko järjestelmällä mahdollista selvittää paras loppuratkaisu:
 - Aukot tietosisällössä
 - Aukot toiminnoissa
- Jos tilanteen suoritus menee läpi, arvioidaan käytettävyyttä:
 - Tehokkuusongelmat eli käyttäjältä vaadittu tarpeeton työ
 - Mekaaninen tehokkuus = turhat toimenpiteet
 - Kognitiivinen tehokkuus = turha miettimistyö ja mielessä pitäminen
 - (Opittavuutta ei arvioida tässä menetelmässä)

Sari A. Laakso (2015)

5.2 Tuloksena käliongelmiä

Aukot tietosisällössä ja toiminnoissa

Jo oikeaa polkua selvitettäessä (vaiheessa 3) ongelmat puuttuvassa tietosisällössä tai toiminnallisuudessa tulevat ilmi:

- Onko tavoite ylipäättään mahdollista saavuttaa järjestelmän avulla, vai puuttuuko tarvittavia tietoja tai toimintoja:
 - Tehtävän suorittamisessa tarvittavan tiedon esille kaivamiseen tarvitaan ulkoista tietolähdettä, kuten toista ohjelmaa tai paperilähdettä, tai muuten suoritus ei etene (esim. Stockmannin myymälän katuosoitteen selvittäminen Reittiopasta varten Googlella; potilaan hoitotietojen etsiminen arkistomapista).
 - Tehtävän suorittamisessa tarvitaan ulkoista apuvälinettä, kuten muistiinpanopaperia tai itse tehtyä Excel-taulukkoa.
 - Tavoitetta ei voida saavuttaa loppuun asti järjestelmän avulla, vaan suoritus katkeaa kesken kokonaan tai epäoptimaalisesta kohdasta (esim. hotellihuoneiden hinnat ja varaustilanteen voi selvittää, mutta huoneen varaaminen on tehtävä puhelimella tai menemällä paikan päälle; ajo-ohjetta hotelliin ei ole lainkaan).
 - Tehtävää ei voi tehdä ollenkaan, ei edes osittain.

Sari A. Laakso (2015)

5.2 Tuloksena käliongelmiä

Tehokkuusongelmat (mekaaninen työ)

- Käytön tehokkuutta heikentävät turhat toimenpiteet, esimerkkejä:
 - Järjestelmä pakottaa käyttämään käsillä olevan tilanteen kannalta turhia toimintoja tai pakottaa navigoimaan turhan mutkan (tai turhien välisivujen) kautta.
 - Toimintoja on käynnisteltävä vuorotellen eri paikoista, mikä aiheuttaa edestakaisen navigoinnin kahden tai useamman näytön välillä.
 - Käyttäjän on syötettävä sama tieto tai valinta kahteen tai useampaan kertaan eri paikkoihin taikka tehtävä samat säädöt monta kertaa, kun ohjelma unohtaa ne välillä.
 - Käyttäjän on syötettävä sellainen tieto, jonka järjestelmä voisi hakea automaattisesti toisesta järjestelmästä.

Sari A. Laakso, Antti Latva-Koivisto (2006)

5.2 Tuloksena käliongelmiä Tehokkuusongelmat (kognitiivinen työ)

- Käytön tehokkuutta heikentävä turha kognitiivinen työ eli 'miettimistyö', esimerkkejä:
 - Käyttäjän on pideltävä mielessään vertailtavia vaihtoehtoja, koska vaihtoehdot on esitetty käyttöliittymässä eri paikoissa eikä niitä ole mahdollista saada samanaikaisesti näkyville.
 - Kun käyttäjä löytää itselleen sopivia vaihtoehtoja (esimerkiksi sopivia hotellivaihtoehtoja Tukholmasta), hänen on yritettävä pitää löytämänsä hyvät vaihtoehdot mielessään sen sijaan, että ohjelma muistaisi ne.
 - Käyttäjän on pideltävä mielessään eri vaihtoehtojen ominaisuuksia, jotka vaikuttavat hänen päätöksentekoonsa, esimerkiksi keväällä järjestettävien seminaarien sisältöjä ja kokoontumisaikoja.
 - Käyttäjän pitää laskea päässä jotakin sen perusteella, mitä näytöllä näkyy (esim. elokuvan päättymisajan laskeminen lisäämällä kesto alkamisaikaan tai ensi viikon torstain päivämäärän laskeminen näytöllä näkyvän tämän päivän päiväyksen perusteella).

Sari A. Laakso, Antti Latva-Koivisto (2006)

5.2 Tuloksena käliongelmiä Opittavuusongelmat jätetään huomiotta

- Simulointitestauksessa ei keskitytä opittavuusongelmiin. Ne kyllä näkyvät polun simuloinnissa 'esteinä', jotka arvioija voisi halutessaan poimia muistiin. Miksi näitä ei kerätä?
 - Ensisijainen syy: Valtaosa alkuperäisistä opittavuusongelmista yleensä häviää samalla, kun tehokkuusongelmat korjataan.
 - Opittavuusongelmista ei ole juuri mitään apua paremman ratkaisun suunnittelussa (jos käyttöliittymässä on myös tehokkuusongelmia).
 - Simulointivaiheessa opittavuusongelmiin keskittyminen on arvioijalle helppoa, mutta se heikentää hänen kykyään paikantaa keskeiset tehokkuusongelmat. Arvioijan tarkkaavaisuus kohdistuu tulosten kannalta hyödyttämiin pikkuseikkoihin, eivätkä keskeiset ilmiöt nouse simuloinnin aikana esille.
- Entäpä jos nimenomaan opittavuusongelmista tarvitaan tietoa?
 - Opittavuusongelmien paikantamiseen on olemassa simulointitestausta luotettavampia menetelmiä, kuten kognitiivinen läpikäynti, käytettävyysläpikäynti ja käytettävyystestaus.

Sari A. Laakso (2015)

Millaisia opittavuusongelmat ovat Näitä arvioija ei poimi

- Opittavuus = Mitkä toimenpiteet tai niiden seuraukset ovat vaikeita keksiä tai ymmärtää? Esimerkkejä:
 - Toimenpiteitä, joita on vaikea keksiä:
 - Kohde, johon liittyvät toimenpiteet eivät ole näkyvissä, esim. kuva tai tekstinpätkä, josta ei voi tajuta, että sitä voisi klikata.
 - Toimintopainikkeen tai linkin harhaanjohtava otsikko, joka...
 - ei vaikuta oikealta, vaikka todellisuudessa on se oikea, tai
 - vaikuttaa houkuttavalta, mutta johtaa väärään paikkaan.
 - Käyttöliittymän komponentin sisältöä huonosti kuvaava otsikko: käyttäjä ei keksi, mitä esimerkiksi syötekenttään pitäisi kirjoittaa ja missä muodossa.
 - Toimenpiteiden seurauksia, joita on vaikea ymmärtää:
 - Ohjelman antama vaikeaselkoinen palaute, josta käyttäjä ei ymmärrä, mitä ohjelma teki, esimerkiksi lähtikö viesti nyt vai ei.
 - Huono virheilmoitus, josta käyttäjä ei ymmärrä, mitä tapahtui tai mikä on ongelmana tai miten ongelmaa pääsee korjaamaan.

Sari A. Laakso (2015)

Muita simulointipohjaisia asiantuntija-arviomenetelmiä

- Kognitiivisella läpikäynnillä (cognitive walkthrough; ks. esim. [Lewis97]) arvioidaan ensisijaisesti jokaisen toimenpideaskelen opittavuutta: keksiikö käyttäjä, mikä toimenpide seuraavaksi pitäisi tehdä. Arvioija simuloi käyttäjän toimenpiteitä vaihe vaiheelta ja vastaa jokaisen toimenpiteen yhteydessä neljään apukysymykseen.
- Myös heuristisessa läpikäynnissä (heuristic walkthrough) [Sears97] arvioija tarkastelee käyttäjän toimenpidesekvenssiä, mutta neljän apukysymyksen lisäksi hän etsii ongelmia käyttämällä Nielsenin tarkistuslistaa [Nielsen94].

Sari A. Laakso, Antti Latva-Koivisto (2006)

Käytettävyystestaus



Käytettävyystestissä (usability test) testikäyttäjille annetaan oikeita työtehtäviä, joista he yrittävät suoriutua järjestelmän avulla. Näin saadaan selville ensisijaisesti käytettävyyssongelmia.

Kirjallisuutta käytettävyystestauksesta: [Nielsen93, luku 6], [Rubin94], [Lauesen05, luvut 1.4 ja 13]

Kuva:
Käytettävyystestausta Helsingin yliopiston tietojenkäsittelytieteen laitoksen Käyttöliittymät-kurssilla keväällä 1999.

Käytettävyystestaus Menetelmä

- Käytettävyystestissä jäljitellään todellista työntekeä: testikäyttäjä etsii sopivaa suoritustapaa, jolla hän saisi testitehtävässä kuvattun tilanteen hoidettua.
 - Testikäyttäjät valitaan järjestelmän todellisesta kohderyhmästä.
 - Testin ohjaaja antaa testikäyttäjälle oikeita käyttötilanteita vastaavia testitehtäviä.
 - Testikäyttäjä suorittaa tehtäviä itsenäisesti parhaaksi katsomallaan tavalla ilman testin ohjaajan apua.
- Testin ohjaaja tai joku toinen testiä seuraava henkilö tekee havaintoja käyttäjän toimenpiteistä ja ajatuksista sekä tulkitsee niistä käyttöliittymän ongelmakohtia.

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitehtävät

Tehtävät todellisia käyttötilanteita

- Käytettävyydestin testitehtävät vastaavat simulointitestauksen testitapauksia, vaikka ne kirjoitetaankin hieman erilaiseen muotoon.
- Testitehtävät ovat realistisia käyttötilanteita, joita järjestelmän oikeille käyttäjille tulee eteen heidän työssään [Rubin94, s. 179].
- Käyttäjälle annetaan testitehtävässä vain tavoite (lähtötilanne). Tehtävänkuvauksessa ei pidä antaa minkäänlaisia vihjeitä toiminnoista, joiden avulla tehtävää suoritetaan [Rubin94, s. 180].

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitehtävät

Tehtävien valinta ja muu aineisto

- Testitehtävien tulisi koostua käyttäjien yleisimmistä tavoitteista ja käyttötilanteista [Rubin94, s. 102].
- Testi kannattaa aloittaa lyhyellä ja suoraviivaisella tehtävällä, jotta käyttäjä kokee saavansa jotain tehdyksi ja pääsee hyvin alkuun [Nielsen93, s. 186]. Monivaiheiset ja hankalat tehtävät kannattaa sijoittaa testin loppuosaan, jotta niitä on mahdollista pudottaa pois, jos aika ei riitä.
- Testitehtävien yhteydessä tulee käyttää todellisuutta vastaavaa aineistoa (dokumentteja, taulukkoja ym.). Epärealistisen yksinkertaisella aineistolla tehty testi ei paljasta kaikkia ongelmia, joita esimerkiksi hyvin laajan aineiston käsittelyssä voi olla.

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitehtävät

Esim: VR:n junalippuautomaatti 1/2

Hyödyllinen tehtävä:

Olet pääsiäislomaa edeltävänä perjantaina Helsingin rautatieasemalla. Ensi keskiviikkona aiot lähteä pääsiäisen viettoon vanhempiesi luokse Pieksämäelle viimeisen luentosi jälkeen, joka päättyy klo 16 keskustassa Porthaniassa. Ajattelit hankkia itsellesi junalipun pääsiäistä varten jo nyt, koska tiedät junien olevan silloin usein täynnä.

Ongelmallisia tehtäviä:

Selvitä, mihin aikaan Pieksämäelle menee junia pääsiäislomien aikoihin. Selvitä myös hinnat sekä opiskelijalipuista että normaalihintaisista aikuisten ja lasten lipuista. Hanki lopuksi lippu johonkin haluamaasi junaan.

Selvitä, montako junaa menee arkisin Helsingistä Pieksämäelle. Kuinka monta niistä on InterCity-junia?

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitehtävät

Esim: VR:n junalippuautomaatti 2/2

Hyödyllinen tehtävä:

Asut Riihimäellä, mutta olet ollut tänään käymässä Helsingissä. Tapasit kavereitasi ja kävitte ostoksilla. Nyt kello on 18.45, kun saavut väsyneenä Helsingin rautatieasemalle. Tiedät, että junia Riihimäelle menee tosi usein, ja ajattelit mennä kotiin seuraavalla sopivalla junalla. Hanki itsellesi junalippu.

Ongelmallisia tehtäviä:

Osta junalippu tänään klo 19.04 Helsingistä Riihimäelle lähtevään junaan.

Osta opiskelijalippu ensimmäiseen klo 18.45 jälkeen Helsingistä Riihimäelle lähtevään junaan.

Selvitä, mikä on nopein juna Helsingistä Riihimäelle arkisin ma-pe. Mitkä ovat lippuhinnat nopeimpaan junaan?

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testikäyttäjät

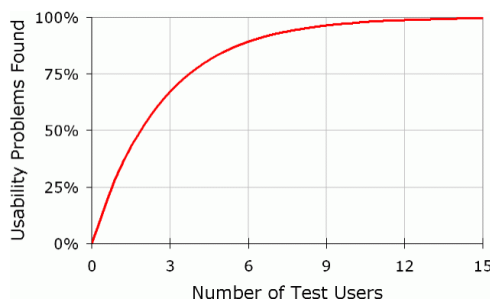
Käyttäjät oikeasta kohderyhmästä

- Testikäyttäjät on tärkeää valita testattavan tuotteen todellisesta kohderyhmästä ([Rubin94, s. 119], [Nielsen93, s. 169]). Väärän kohderyhmän käyttäjillä testaaminen voi johtaa siihen, ettei todellisen käytön kannalta keskeisimpiä ongelmia saada selville.
 - Softan kehittäjän 25-vuotias nörttikaveri voi nauttien selviytyä käsittämättömästä toimintoviidakosta, jossa "luodaan hierarkkisia profiileja" ja "säädetään IMAP-asetuksia", mutta tuotteen ensisijaiseen kohderyhmään kuuluva käyttäjä ei välttämättä onnistu lukemaan sähköpostejaan ollenkaan.
 - Kohderyhmään kuulumattomalta testikäyttäjältä voi puuttua työn tekemiseen tarvittavia tietoja ja taitoja. Esimerkiksi kuva-arkiston käytettävyydesteihin tarvitaan testikäyttäjiksi kuvatoimittajia, jotka tietävät, millä perusteilla kuvia todellisessa työssä lehteen valitaan.

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testikäyttäjät

Käyttäjien lukumäärä



Lähde: [Nielsen00]

Jos testaat yhtä käyttöliittymää, 3-5 käyttäjää on yleensä riittävä määrä.

Mitä useammalla käyttäjällä testaat, sitä useammin samat ongelmat toistuvat. Testaukseen käytetyt lisäresurssit tuovat enää vain vähän lisää hyötyä.

Sari A. Laakso, Antti Latva-Koivisto (2006)

Muut testivalmistelut

Testitila ja pilottitesti

- Erillistä käytettävyysslaboratoriota ei tarvita, vaan testi voidaan järjestää esim. tavallisessa työhuoneessa [Nielsen93, s. 200].
- Testihuone lavastetaan oleellisilta osin todellista käyttöympäristöä vastaavaksi. Esimerkiksi matkatoimistossa käytettävän järjestelmän testauksessa voisi olla hyödyllistä lavastaa jotakin seuraavanlaista:
 - asiakaspalvelutiski ja asiakkaiden käyntejä,
 - puhelin ja asiakkaiden puhelinsoittoja ja
 - matkatoimistossa käytettäviä paperiesitteitä ja muuta materiaalia.
- Ennen varsinaisia testejä kannattaa järjestää ns. pilottitesti eli harjoittelutesti, jonka tuloksia ei oteta huomioon varsinaisissa testituloksissa. Pilottitestin tarkoituksena on harjoitella testin läpivientiä ja korjata jäljellä olevat ongelmat testitehtävistä, testin ohjauksesta ja testattavan prototyypin toiminnasta [Nielsen93, s. 174-175].

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitilanteen ohjaaminen

Ennen testin aloittamista

- Yritä saada testin alku rennoksi ja luontevaksi. Käyttäjät ovat usein hermostuneita aloittaessaan testin, kun he eivät tiedä, mitä tuleman pitää. Usein he kokevat myös valtavia suorituspaineita, joita tarkkailun kohteena oleminen vielä lisää entisestään [Nielsen93, s. 181].
- Kerro testikäyttäjälle ennen testin aloittamista seuraavat toimintaperiaatteet [Nielsen93, s. 188; Rubin94, s. 148-149]:
 - *Emme testaa sinua, vaan tätä ohjelmaa. Tarkoituksena on löytää ohjelmasta niin paljon ongelmia kuin mahdollista.*
 - *Jos haluat, voit lopettaa testin milloin tahansa.*
 - *Jos testi videokuvataan: Emme ole kuvaamassa sinua, vaan kamera on sitä varten, jotta muistaisimme jälkepäin testitilanteesta esille tulleet asiat.*
 - *Pyytäisin ajattelemaan ääneen, mitä olet tekemässä [Nielsen93, s. 195].*

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitilanteen ohjaaminen

Testitilanteen alussa

- Kerro käyttäjälle lyhyesti, millaisesta ohjelmasta testissä on kyse, ja kuvaile testitehtävien taustatilanne lyhyesti:
 - Järjestelmän tarkoitus noin yhdellä lauseella, esim. *“Tällä ohjelmalla käsitellään kodinkoneliikkeeseen tulevia asiakasvalituksia.”*
 - Kuka käyttäjä on ja missä hän kuvitellusti on, esim. *“Olet juuri tullut ensimmäisenä työpäivänäsi kodinkoneliikkeen valitustiskille töihin. Jotkut asiakkaasi tulevat tähän tiskille esittämään valituksensa ja osa soittaa sinulle puhelimella.”*
 - Jos testiympäristöön kuuluu esim. puhelin, kerro siitä käyttäjälle ennen testin alkua, esim. *“Tuo puhelin on tässä testissä oma työpuhelimesi. Jos se soi, voit vastata siihen omalla nimelläsi.”*
- Tarkoituksena on saada käyttäjän sovellusalueeseen ja testiasetelmaan (mutta ei itse käyttöliittymään!) liittyvä tietämys todellista tilannetta vastaavaksi.
- Järjestelmän toimintaa ei pidä esitellä käyttäjälle etukäteen, jos käyttäjä ei tositilanteessakaan saisi koulutusta ennen käyttöä.

Sari A. Laakso (2015)

Testitilanteen ohjaaminen

Testitehtävien antaminen käyttäjälle

- Anna testitehtävät käyttäjälle yksi kerrallaan.
- Kuvaile testitehtävän sisältämä esimerkkitalanne käyttäjälle suullisesti vapaamuotoisesti.
- Yleensä testitehtävä kannattaa antaa käyttäjälle myös paperilapulla [Nielsen93, s. 186], jottei hänen tarvitsisi pitää mielessään tehtävässä annettuja tietoja.
- Siirry tehtävästä toiseen huomaamattomasti kuvailemalla uutta tilannetta. Älä kommentoi käyttäjän suoriutumista tehtävästä [Nielsen93, s. 190], koska käyttäjän kykyjä ei olla nyt testaamassa.

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitilanteen ohjaaminen Ulkopuoliset tarkkailijat

- Testitilassa voi olla testikäyttäjän ja testin ohjaajan lisäksi muita henkilöitä (esim. 1-2 henkilöä tarkkailemassa tai tekemässä muistiinpanoja), mutta he eivät saa häiritä testiä tai osallistua testin kulkuun [Nielsen93, s. 190].
- Yleensä ulkopuolisten tarkkailijoiden on parempi seurata testiä testihuoneen ulkopuolella esim. TV-ruudulta tai yksisuuntaisen peilin läpi (jos testi järjestetään käytettävyysslaboratoriossa).

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitilanteen ohjaaminen Neuvomisen välttäminen

- Älä neuvo tai auta käyttäjää tehtävän tekemisessä äläkä anna minkäänlaisia vihjeitä tehtävän suorittamisesta [Nielsen93, s. 183; Rubin94, s. 216, 222].
 - Jos käyttäjä esimerkiksi keksii uusia tai yllättäviä toimintatapoja, anna hänen tehdä toimenpiteet itse loppuun asti.
 - Älä 'vedätä' tai johdattele ennalta suunniteltuun suoritustapaan.
 - Älä näytä yllätyneeltä tai ala hämmästellä tilannetta, vaikka käyttäjä tekisi jotain täysin odottamatonta [Rubin94, s. 220].
 - Jos käyttäjä kysyy sinulta suoraan ohjelman toiminnasta, esim. "Tulostaako tämä, jos painan nyt tästä?", esitä vastakysymys [Nielsen93, s. 196], esimerkiksi "Mitä arvelisit? Miltä se sinusta vaikuttaa? Voit kokeilla vapaasti kaikkea."
- Testin ohjaajalla on monesti suuri houkutus käyttäjän neuvomiseen testin aikana, mutta neuvominen pilaa helposti testin tulokset (vrt. [Nielsen93, s. 183]).

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitulokset

Ongelmien havainnointi

- Käyttöliittymäongelmia löytyy hyvin, kun testaaja lähtee siitä oletuksesta, että testikäyttäjä toimii aina järkevästi. Kaikki testin aikana havaitut käyttäjän virheet ja ongelmat tehtävissä johtuvat lähtökohtaisesti käyttöliittymän ongelmista ja puutteista, eivätkä käyttäjästä. Tavoitteena on löytää tuotteesta syy käyttäjän kohtaamille ongelmille [Rubin94, s. 276].
- Käyttäjän kohtaamien ongelmien syitä eli käyttöliittymän vikoja voidaan päätellä
 - käyttäjän tekemistä toimenpiteistä,
 - käyttäjän reaktioista ja
 - ääneenajattelusta [Nielsen93, s. 195].

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitulokset

Toimenpiteiden taltioiminen 1/2

- Yritä saada toimenpiteet talteen tekemällä heti testin aikana käsin muistiinpanoja.
 - Käyttösekvenssin voi kirjoittaa muistiin yksinkin, mutta sitä on kuormittavaa tehdä testin ohjaamisen ohella.
 - Monesti on käytännöllistä pyytää työkaveria avuksi ohjaamaan testiä tai tekemään muistiinpanoja. Hyvien muistiinpanojen tekeminen on vaativampaa kuin testin ohjaus.
 - Videokamera on hyvä lähinnä varajärjestelynä: videonauhalla voit jälkepäin tarkistaa käyttäjän tekemisiä tai puheita, jos et ehtinyt nähdä, mitä käyttäjä teki jossakin tärkeässä kohdassa.
- Merkitse muistiin ensisijaisesti se, mitä käyttäjä oikeasti teki (todelliset tapahtumat). Testin aikana tehdyt tulkinnat voivat osoittautua myöhemmin virheellisiksi tai hyödyttömiksi. Jos alkuperäistä tapahtumadataa ei ole tallessa, tulkintoja ei voida myöhemmin muuttaa vastaamaan testissä ilmennyttä uutta dataa.
- Vrt. Lauesenin esimerkki käytettävyydestin muistiinpanoista [Lauesen05, s. 440].

Sari A. Laakso, Antti Latva-Koivisto (2006)

Testitulokset

Toimenpiteiden taltioiminen 2/2

Esimerkki YTV:n Reittioppaan käytettävyydestin muistiinpanoista:

- Ote käyttösekvenssistä ja käyttäjän ääneenajattelusta:
...Käyttäjä painoi Hae. Avasi Mistä-pudotuslistan, jossa annettu kaksi eri Unikkotietä. "Ahaa, kaks Unikkotietä, Hesassa ja Vantaalla. Mä vaihdan tuoksi Vantaan Unikkotieksi." Valitsi Vantaan Unikkotien, klikkasi [Vaihda](#), järjestelmä tyhjensi kentän. "Mitä ihmettä, mitä nyt tapahtu?" Kirjoittaa uudelleen Unikkotie 6, painaa Hae, valitsee Vantaan Unikkotien, klikkaa Hae.

Sari A. Laakso (2006)

Testitulokset

Ongelmien jäsentäminen 1/2

- Käyttöliittymän ongelmakohdat nimetään kuvaavasti, ja ongelmat perustellaan käyttäjän toimenpiteistä ja ääneenajattelusta tehdyillä havainnoilla.
- Esimerkki jäsentelystä:

Ongelma 1: <Kuvaava otsikko>

Mitä käyttäjä pyrki saamaan aikaan (tavoite/alitavoite)?

Mitä käyttäjä teki (toimenpiteet) ja mitä hän sanoi (ääneenajattelu)?

Mitä käyttäjän olisi pitänyt tehdä?

Mikä ongelma tästä seurasi käyttäjälle?

Ongelman syy käyttöliittymässä?

Sari A. Laakso (2015)

Testitulokset

Ongelmien jäsentäminen 2/2

Ongelma 1:

Käyttäjä etsi ruokapaikkoja Neste 24 h -linkin rekisteröitymisen takaa.

Mitä käyttäjä pyrki saamaan aikaan (tavoite/alitavoite)? Esim. käyttäjä yritti selvittää, miltä Nesteen asemilta ylipäätään saa ruokaa.

Mitä käyttäjä teki (toimenpiteet) ja mitä hän sanoi (ääneenajattelu)? Käyttäjä valitsi yläreunan päävalikosta *Neste 24 h* -linkin ja sanoi: "Täällä on varmaan kaikki palvelut." Mutisi tyytymättömänä: "Pitääkö tännekin rekisteröityä." Jatko valitsemalla *Rekisteröidy tästä*.

Mitä käyttäjän olisi pitänyt tehdä? Hänen olisi pitänyt valita oikeasta reunasta *Karttapalvelu* tai *Neste Oil -asemahaku* taikka lähteä liikkeelle yläreunan päävalikostavalitsemalla *Neste Oil -asemaverkosto*.

Mikä ongelma tästä seurasi käyttäjälle? Hän yritti pitkään (yli 10 minuuttia) selvittää, miten hän saisi käyttäjätunnuksen Nesteen sivustolle, eikä saanut mitään tietoa ruokaa tarjoavista huoltoasemista. Hän unohti alkuperäisen tavoitteensa ja ryhtyi selvittämään rekisteröitymistä.

Ongelman syy käyttöliittymässä? *Neste 24 h* -linkin otsikko vaikutti kuvaavammalta asemien ruokailupalvelujen selvittämiseen kuin mikään linkeistä *Karttapalvelut*, *Neste Oil -asemahaku* tai *Neste Oil -asemaverkosto*. <- **HUOMAA:** *Kun arvioit ongelman syytä käyttöliittymässä, älä lähde siitä, että ohjeita puuttuu tai että käyttöliittymään pitäisi lisätä ohjetekstejä. Yritä löytää ongelman varsinainen syy (tai syitä).*

Sari A. Laakso (2015)

Testitulokset

Testissä näkyviä oireita ongelmista

- Mahdollisia oireita käyttöliittymän ongelmista:
 - Käyttäjä ei onnistu tekemään testitehtävää.
 - Käyttäjä luulee tehneensä tehtävän kokonaan, mutta todellisuudessa suoritus on keskeneräinen tai lopputulos on väärä.
 - Käyttäjä tekee suorituksessa virheen, kuten syöttää dataa väärään kenttään [Rubin94, s. 276], esim. etunimen sukunimikenttään, tai hävittää erehdyksessä aiemmin luomaansa dataa, esim. unohtaa tallentaa tiedoston.
 - Käyttäjä tekee epäoptimaalisia toimenpiteitä, jotka poikkeavat odotetusta oikeasta suoritustavasta [vrt. Rubin94, s. 276].
 - Käyttäjä käynnistää turhaan väärän toiminnon, avaa väärän ikkunan tai menee väärälle sivulle, mutta huomaa virheensä saman tien ja korjaa sen (esim. sulkee ikkunan, painaa *Back*).
 - Käyttäjä epäröi pitkään jossakin vaiheessa tehtävän suoritusta, mutta suorittaa lopulta tehtävän oikein. (Pitkän epäröinnin aikana käyttäjää kannattaa muistuttaa ääneenajattelusta.)
 - Käyttäjä valittaa ääneen jostakin ongelmasta.

Antti Latva-Koivisto, Sari A. Laakso (2006)

Testitulokset

Ongelmien vakavuuden arviointi

1. Puuttuva toiminnallisuus: Käyttöliittymällä ei pysty suorittamaan testitehtävää. Esim: *Käyttäjä yritti tulostaa alennuskoodit tehdäkseen niihin omia merkintöjään, mutta järjestelmästä ei voi tulostaa koodeja.*
2. Epäonnistunut tehtävän suoritus: Testikäyttäjä ei pystynyt suorittamaan tehtävää itsenäisesti tai luuli (virheellisesti) tehneensä tehtävän jo. Esim: *Käyttäjä luuli tehneensä tehtävän ja tuloksen olevan tallessa, mutta hänen olisi pitänyt vielä painaa Update-painiketta.*
3. Selvästi häiritsevä ongelma: Käyttäjä ei osannut toimia optimaalisella tavalla tai hän valitti suullisesti järjestelmän olevan hankala tai kömpelö. Esim: *Käyttäjä sanoi, että on täysin älytöntä kulkea kuuden näytön läpi, jotta saisi täytettyä kymmenen kenttää.*
4. Melko häiritsevä ongelma: Käyttäjä sai tehtävän suoritettua pitkällisten yritysten jälkeen. Esim: *Käyttäjä yritti haun käynnistämistä monella eri tavalla ennen kuin keksi näytöltä opasteen ja painoi F10.*
5. Vähäinen ongelma: Käyttäjä löysi ratkaisun muutaman lyhyen yrityksen jälkeen. Esim: *Käyttäjä yritti käynnistää haun ensin enterillä, mutta huomasi sitten painaa F10-näppäintä.*

[Lauesen05, s. 12-13]

Antti Latva-Koivisto (2006)

Kustannustehokas testaus

- Videoidut käytettävyytestit suurilla käyttäjämäärillä ovat kalliita ja vievät paljon aikaa. Muu projekti voi ehtiä edetä jo pitkälle, kun testejä vasta analysoidaan.
- Monesti yhden laajan testin sijaan kannattaa kerätä nopeasti tietoa pahimmista ongelmista järjestämällä monta kevyttä ja edullista testiä. Testien välillä korjataan pahimmat ongelmat.
 - Järjestä testejä esim. tavallisessa työhuoneessa tai neukkarissa.
 - Valitse 3-4 todellista käyttäjää, mutta älä koodaaja-työkavereita.
 - Laadi testitehtäviksi aitoja tilannekuvauksia.
 - Tee ongelmakohdista muistiinpanot testin kuluessa, ja pura muistiinpanot heti testin jälkeen. Voit hoitaa testit yksinkin, mutta jos mahdollista, pyydä työkaveria avuksi joko muistiinpanojen tekijäksi tai testin ohjaajaksi.

Sari A. Laakso (2006)

Käyttötilanteet

Käyttötilanteita tarvitaan sekä testitapauksiksi että käyttöliittymän suunnittelun syötteeksi.

Testitapauksena oleva käyttötilanne kuvaa käyttötilanteen lähtöasetelman ja motiivit mutta ei vielä järjestelmän käyttöä (mitä käyttäjä tekee ja miten järjestelmä reagoi), koska juuri tätä on tarkoitus ryhtyä testaamaan. Käyttötilanne virittää aina jonkin ongelman, jonka ratkaisemisessa tietojärjestelmän on tarkoitus auttaa.

Käyttötilanteet testitapauksina Esimerkki: Matkahuolto ja VR

- Hyvä testitapaus on riippumaton testattavasta järjestelmästä: testitapauksessa ei kuvata vielä järjestelmän käyttöön liittyviä tehtäviä tai toimenpiteitä, vaan käyttöä edeltävää tilanneasetelmaa.
- Seuraavassa esimerkissä testitapaukseksi on valittu todellinen tilanne, jossa Marko oli menossa kummipoikansa syntymäpäiville Turkuun ja yritti selvittää, miten pääsisi sinne.

Käyttötilanne: Kummipojan syntymäpäiville Turkuun

Marko, 27-vuotias ravintolakokki, on maanantai-iltana kotonaan Helsingin Kalliossa osoitteessa Kolmas linja 23. Hän on menossa ensi lauantaina 4-vuotiaan kummipoikansa syntymäpäiväjuhliin Turkuun. Juhlat alkavat klo 14 kummipojan kotona Rauhankadulla, joka sijaitsee Turun keskustassa rautatie- ja linja-autoaseman lähellä.

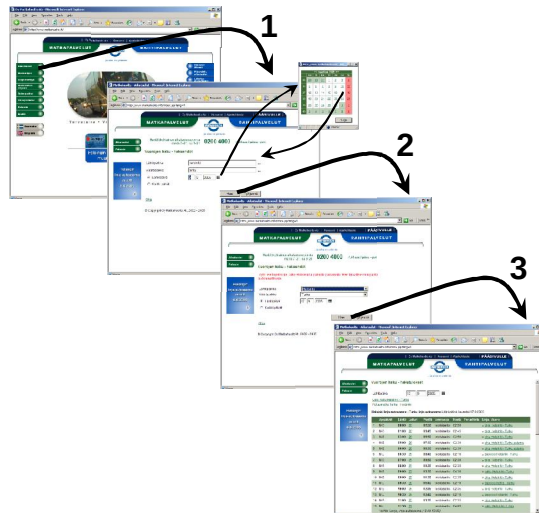
- Testataan nyt käyttöä simuloimalla, miten hyvin Matkahuollon ja VR:n sivustot tukevat Markon tilannetta.

Sari A. Laakso, Antti Latva-Koivisto (2006)

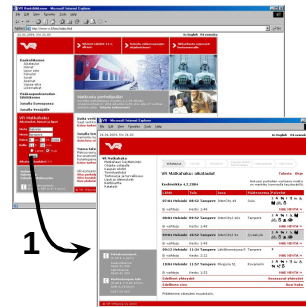
VR:n ja Matkahuollon sivustojen simulointitestausta

Sivusiirtymät menomatkan aikataulujen selviämiseen asti

Matkahuolto



VR



Antti Latva-Koivisto, Sari A. Laakso (2006)

Käyttötilanteet testitapauksina

Lähtötilanne, ei järjestelmän käyttöä

- Lähtötilanne pysyy samana, vaikka tehtävän suoritustapaa ja käyttöliittymää muutetaan (vrt. brief scenario, [Hackos98, s. 324]). Käyttötilanne...
 - kuvailee tilanteen, joka käyttäjien pitää pystyä hoitamaan käyttöliittymän avulla, mutta
 - ei kuvaa sitä, miten käyttäjät tekevät tämän tehtävän nykyisellä käyttöliittymällä (mitä toimintoja käynnistävät, mitä tietoja katsovat käyttöliittymästä, missä järjestyksessä jne.).

Sari A. Laakso, Antti Latva-Koivisto (2006)

Käyttötilanteet testitapauksina

Lähtötilanne, ei järjestelmän käyttöä

Lähtötilanne:

Petteri on yliopiston päärakennuksessa viimeistelemässä konferenssipaperiaan kahden työkaverinsa kanssa. On jo ilta klo 21.15, ja kaikilla alkaa olla hirveä nälkä. Kirjoittaminen on vielä ihan kesken, ja sen kanssa näyttää menevän myöhään. Kellään ei ole mukana mitään syötävää.

Ei järjestelmän käyttöä:

Petteri avaa hampurilaisravintolan tilauspalvelun ja valitsee 6 megaburgeria ja kolmet isot ranskalaiset. Sitten hän syöttää osoitteen ja painaa tilauspainiketta. Näytön alareunaan ilmestyy viesti: "Tilaus lähetetty, saapuu puolen tunnin kuluttua."

- Testitapauksen sisältämä lähtötilannekuvaus on syöte testaukselle.
- Vasta testauksen aikana selvitetään, miten käyttäjä pääsee tavoitteeseensa käyttöliittymän avulla. Käytön kuvaus on siis testauksen tuotos.

Sari A. Laakso, Antti Latva-Koivisto (2006)

GDD:n käyttötilanteet

Ei päätöksiä ja kiinnitettyjä ratkaisuja

Lähtötilanne, jonka seassa on käyttäjän tekemiä päätöksiä ja kiinnitettyjä ratkaisuja (alleviivattu):

Helsinkiläinen kardiologi Terhi on menossa kuuntelemaan ensi viikon torstaina Turussa klo 12 alkavaa Urheilijan sydän –seminaaria. Hän on katsonut Matkahuollon verkkopalvelusta bussien aikatauluja ja ostanut liput. Koska seminaari alkaa klo 12, hän valitsi tarpeeksi varhaisen bussin, joka lähtee Helsingistä jo klo 8. Seminaarin viimeinen esitys pidetään klo 17.15-17.45, joten hän päätyi klo 18.30 lähtevään bussiin. Lopuksi hän tilasi ostamansa liput sähköpostiinsa ja tulosti ne myös paperille varmuuden vuoksi. Hän tietää, että seminaaripaikka on keskustassa osoitteessa Brahenkatu 10 noin viiden minuutin kävelymatkan päässä linja-autoasemasta.

Miten korjataan:

- Siirrä tilanteen aktivoitumishetki niin varhaiseksi, ettei käyttäjä ole vielä tehnyt päätöksentekoon tähtäviä toimenpiteitä (esim. "on katsonut bussien aikatauluja", "valinnut klo 6.42 lähtevän bussin").
- Poista kiinnitetty ratkaisut (esim. lippujen tilaaminen sähköpostiin).

Sari A. Laakso (2015)

GDD:n käyttötilanteet

Käyttäjän tavoite (motiivi)

- GDD:n vaatimaan käyttötilannekuvaukseen sisältyy käyttäjän tavoite (motiivi): mitä hän on yrittämässä tehdä ja miksi?
 - Testauksen kannalta ei ole juurikaan merkitystä, mikä motiivi tilanteessa on, kunhan se on realistinen ja riittävän tyypillinen.
 - Jos motiivi puuttuu, testaaaja ei pysty ottamaan huomioon tilanteeseen tyypillisesti vaikuttavia tekijöitä. Tästä seuraa, ettei testauksella saada esiin kaikkia keskeisiä, todellisissa tilanteissa eteen tulevia ongelmia. Esimerkki:

Käyttötilanne: Ensi viikon torstaina Turkuun
Helsinkiläinen kardiologi Terhi on menossa ensi viikon torstaina Turkuun. Perillä pitää olla klo 12, ja paluu on klo 18.
 - Puuttuvan motiivin takia aikatauluhaun testauksesta menisi ongelmitta läpi huono käyttöliittymäratkaisu, joka näyttäisi kerralla vain ensimmäisen paluubussin (esim. klo 18 lähtevän), vaikka monissa todellisissa tilanteissa Terhin kannattaisikin valita myöhemmin (esim. klo 18.30 tai klo 19) lähtevä bussi.

Antti Latva-Koivisto, Sari A. Laakso (2006)

GDD:n käyttötilanteet

"Käyttäjä haluaa" -ongelma

- Testitapausta laatiessa kannattaa välttää ylimalkaista ilmausta "Käyttäjä haluaa/haluaisi/tahtoo", koska tällöin testitapauksen motiivi jää epäselväksi eivätkä kaikki todellisen tilanteen ongelmat tule esille (vrt. edellinen kalvosivu).
- Koska testitapauksesta ei selviä, mihin haluaminen perustuu, osa "haluamisesta" voi olla epärealistista tai koko testitapausta voi osoittautua virheelliseksi tai niin epätyypilliseksi, ettei sillä kannata testata.
- Haluamisen takana olevan informaation saat esille yleensä kysymällä "miksi".

Ei käyttäjän haluamista:
Helsinkiläinen kardiologi Terhi on menossa ensi viikon torstaina Turkuun. Hän haluaa olla perillä klo 12 ja haluaa lähteä takaisin klo 18.

Kuvaa tilannetta ja syitä haluamisen takana:
Helsinkiläinen kardiologi Terhi on menossa ensi viikon torstaina kuuntelemaan Turussa klo 12 alkavaa Urheilijan sydän – seminaaria, joka alkaa liikuntalääketieteen erikoislääkärin tilannekatsauksella (klo 12-13). Päivän viimeinen esitys pidetään klo 17.15-17.45...

Sari A. Laakso (2006)

Käyttötilanteiden kerääminen haastatteluilla

- GDD:ssä käyttäjähaastattelujen tarkoituksena on kerätä pelkästään konkreettisia tilanneasetelmia, joissa näkyy motiivi ja käyttäjän keskeinen tietämys tai puuttuva tietämys. (Samalla tavalla kerätyt tilanteet sopivat myös käytettävyydestäuksen tai muiden arviointimenetelmien syötteeksi.)
- Keskeisin vaikeus haastatteluissa: käyttäjät yleistävät asioita.
 - He yrittävät puhua kaikkien käyttäjien puolesta sen sijaan, että he kertoisivat omista kokemuksistaan.
 - He tekevät yhteenvetoja aiemmin sattuneista tilanteista tai kertovat niistä niin yleisellä tasolla, ettei kertomuksista saa aikaan käyttötilannekuvauksia.
- Haastattelijan tulee houkuttaa käyttäjä konkretisoimaan:
 - Kohdenna kysymys konkreettiseen tapahtumaan: "Milloin sinulle viimeksi kävi niin? Mitä silloin tapahtui? ..."
 - Jos haastateltava ei anna esimerkkejä, kannattaa itse laatia nopeasti esimerkki ja pyytää häntä korjaamaan keksittyä esimerkkiä.

Sari A. Laakso (2015)

GDD:n käyttäjähaastattelut Käyttäjä välttelee esimerkkejä

Haastattelija kysyy liian yleisesti eikä saa mitään selville:

- *Käytätkö usein yliopiston verkkosivuja? Mihin tarkoitukseen?*
 - "Kyllä minä oikeastaan aika usein katson erilaisia ammattiasioita yliopiston sivuilta."
- *Millaisia ammattiasioita?*
 - "Aivan kaikenlaisia."
- *Antaisitko esimerkin?*
 - "No siis ne voivat olla aivan mitä tahansa."
- *Kertoisitko yhden?*
 - "Se riippuu ihan tilanteesta."
- *Millä tavalla se riippuu tilanteesta?*
 - "Jollain viikolla voi olla kiire ja silloin en välttämättä käytä ollenkaan... Joskus kun on vähän tyhjää aikaa, rupean selvittämään sitten jotain käytännön asiaa."
- *Millaista käytännön asiaa?*
 - "Sellaista, mikä olisi hyvä hoitaa mutta on jäänyt..."

Sari A. Laakso (2015)

GDD:n käyttäjähaastattelut

Haastatteli houkuttaa konkretiaan

Haastatteli kohdistaa kysymyksen äskettäiseen hetkeen:

- Käytätkö usein yliopiston verkkosivuja? Mihin tarkoitukseen?
 - "Kyllä minä oikeastaan aika usein katson erilaisia ammattiasioita yliopiston sivuilta."
- Milloin viimeksi katsoit jotakin ammattiasiaa verkosta?
 - "Taisi olla eilen..."
- Mitä katsoit?
 - "Henkilöstöhallinnon sivuja."
- Mitä etsit sieltä?
 - "Että mitä liitteitä virkahakemuksessa pitää olla ja milloin on deadline."
- Miksi?
 - "No joo, siis... Laitoksella on assistentin virka auki ja olen miettinyt, että voisin hakea..." jne. [Nyt haastatteli pääsee selvittämään ensimmäistä konkreettista tilannetta.]

Sari A. Laakso (2015)

GDD:n käyttäjähaastattelut

Unicafe-esimerkki

- Kun olet päässyt kiinni esimerkitilanteeseen ("Milloin viimeksi...?"), jatka selvittämällä, miksi ko. henkilö teki niin kuin teki (motiivi).
- Esimerkki: Unicafe-käyttäjähaastattelu verkkosivujen kehittämiseksi
 - Milloin viimeksi olet syönyt Unicafessa? Missä? <- KIINNI TILANTEeseen
 - Miksi menit sinne? <- ALETAAN HAKEA MOTIIVIA
 - Mitä teit juuri sitä ennen? Missä olit? Miksi? <- MOTIIVIN HAKU JATKUU
 - Minne olit sen jälkeen menossa? Miksi?
- Seuraavalla sivulla on esimerkki käyttäjältä selvitetystä Unicafe-tilanteesta (alkuperäisen tilanteen selvittäjä on Joonas Gustafsson, 2004; editoitu luentomateriaaliin sopivaksi).
 - Huomaa, että todellisessa tilanteessa kyseinen haastateltava oli vain yksinkertaisesti kävellyt Ylioppilasaukion Unicafehen ja katsonut siellä ruokalistaa ja valinnut sieltä parhaan tai vähiten huonon vaihtoehdon.
 - Käyttötilannekuvauksessa ei kiinnitetä toteumaa ja juututa siihen, esim. käyttäjä tietää Ylioppilasaukion ja menee sinne. Käyttöliittymän suunnittelussa GDD:llä selvitetään ensiksi, minne käyttäjän tässä tilanteessa olisi oikeasti kannattanut mennä (saattaa olla eri kuin Ylioppilasaukio).

Sari A. Laakso (2015)

Erilaisia suunnitteluprosesseja Lauesen, Carroll ja Cooper

- Hyvien käyttöliittymäratkaisujen syntymistä ja systemaattista luomista on tutkittu todella vähän. Tällä hetkellä kirjallisuudessa kuvattuja menetelmiä ovat lähinnä seuraavat:
 - Søren Lauesenin järjestelmällisesti etenevässä *Virtual Windows* -menetelmässä [Lauesen01, Lauesen05] käyttöliittymä suunnitellaan ensin pelkän tietosisällön avulla ja toiminnallisuus lisätään näyttökuviin vasta suunnittelun loppuvaiheessa.
 - *Scenario-Based Design* [Carroll00] -menetelmässä käyttöliittymää suunnitellaan kirjoittamalla tekstimuotoisia skenaarioita eli tarinoita siitä, miten kuviteltu käyttäjä toimisi järjestelmän kanssa.
 - Alan Cooperin *Goal-Directed Design* [Cooper03] -menetelmässä kuvataan tulevia käyttäjiä ja heidän korkean tason tavoitteitaan. Sen jälkeen näiden käyttäjien näkökulmasta kirjoitetaan tekstimuotoisia käyttöskenaarioita, joiden sisältämät tiedot ja toiminnot lopulta organisoidaan käyttöliittymän näkymiksi ja komponenteiksi.

Sari A. Laakso, Antti Latva-Koivisto (2006)

Erilaisia suunnitteluprosesseja Lauesenin *Virtual Windows*

- Lauesenin *Virtual Windows* -menetelmässä käyttöliittymä suunnitellaan ensin pelkän tietosisällön avulla, ja toiminnot lisätään vasta suunnittelun loppuvaiheessa.
 - Ensin selvitetään käyttäjän tehtävät (*tasks*), jotka järjestelmällä pitäisi voida tehdä, ja niistä kirjoitetaan tekstimuotoiset kuvaukset.
 - Sitten tietosisältö jaetaan ns. virtuaali-ikkunoihin simuloimalla käyttäjän tehtäviä ja soveltamalla Lauesenin antamia suunnittelusääntöjä. Tietosisältöä voidaan katsoa tässä vaiheessa myös esimerkiksi tietomallin kuvauksesta.
 - Virtuaali-ikkunoista piirretään visualisoituja oikean näköisiä näyttökuvia. Näissä graafisissa virtuaali-ikkunoissa ei kuitenkaan ole vielä lainkaan toimintoja eikä niitä ole sovitettu mahtumaan lopulliseen näyttökokoon (voivat olla liian pieniä tai isoja näytölle).
 - Lopuksi virtuaali-ikkunoihin lisätään toiminnot ja virtuaali-ikkunat sovitetaan lopulliseen näyttökokoon. Tämä tehdään taas simuloimalla käyttäjän tehtäviä.

Antti Latva-Koivisto, Sari A. Laakso (2006)

Erilaisia suunnitteluprosesseja Carrollin skenaariomenetelmä

- Carrollin *skenaariomenetelmä*ssä [Carroll00] käyttöliittymää suunnitellaan ensisijaisesti tekstimuodossa, ja siirtyminen tekstiskenaarioista näyttökuviin jää osin avoimeksi.
 - Aluksi tehdään skenaarioiden laatimisen pohjaksi kenttätutkimusta: käyttäjien haastatteluja ja heidän nykyisen toimintansa tarkkailua.
 - Tekstimuotoisten skenaarioiden avulla tehdään rinnan käyttöliittymäsuunnittelua ja työnkulkujen uudelleensuunnittelua. Käyttäjän työnkulun etenemistä ja käyttöliittymän toimintalogiikkaa ideoidaan ja kuvataan skenaarioina, joita kirjoitetaan moneen kertaan (iteratiivinen suunnittelu). Käyttöliittymän tietosisältöä tai toiminnallisuutta ei systemaattisesti visualisoida näyttökuvina, kuten Lauesenin prosessissa, mutta apuna voidaan käyttää kuviakin.
 - Kun skenaarioiden kirjoittamisen yhteydessä tulee esille erilaisia suunnitteluratkaisuja, niitä verrataan ns. väittämillä (*claims*). Väittämiin kirjataan ratkaisuvaihtoehtojen hyvät ja huonot puolet, jotta niitä voidaan verrata keskenään.

Sari A. Laakso, Antti Latva-Koivisto (2006)

Erilaisia suunnitteluprosesseja Cooperin *Goal-Directed Design*

- Cooperin *Goal-Directed Design* -menetelmässä [Cooper03, s. 16-20, 39-90] kuvataan tulevia käyttäjiä ja heidän tavoitteitaan sekä laaditaan keskeisimmistä tilanteista skenaariokuvauksia.
 - Aluksi tehdään tulevista käyttäjistä ns. persoonakuvauksia mm. käyttäjähaastattelujen ja -tarkkailujen perusteella. Jokaiselle persoonalle selvitetään/luodaan korkean tason tavoitteet.
 - Seuraavaksi laaditaan käyttöskenaarioita (*context scenarios*) hieman vastaavaan tapaan kuin Carrollinkin menetelmässä. Skenaariossa kuvataan käyttäjän kannalta mahdollisimman hyvä toteuma tulevalla järjestelmällä, mutta se myös kiinnittää jo karkealla tasolla uuden järjestelmän toimintoja ja käyttöliittymän suunnitteluratkaisuja.
 - Skenaarioista tulkitaan käyttöliittymässä tarvittavat toiminnot ja tiedot. Sen jälkeen ne organisoidaan käyttöliittymän näkymiksi ja paneeleiksi "top-down" eli aloittaen karkeista näkymistä ja siirtyen vähitellen kohti näytön yksityiskohtia.

Antti Latva-Koivisto, Sari A. Laakso (2006)

Suunnitteluprosessin valinta

Lopputulos testataan käyttötilanteilla

- Kaikille suunnittelumenetelmille yhteistä on se, että niiden tuottama lopputulos aina viime kädessä testataan todellisilla käyttötilanteilla, kun järjestelmä otetaan käyttöön ja käyttäjät ryhtyvät ensimmäistä kertaa tekemään sillä todellista työtään.
- Kaikki edellä kuvatut prosessit ottavat suunnittelussa huomioon käyttäjien työtehtävät tai käyttäjille eteen tulevat käyttötilanteet.
- (Käyttäjien työnkulkuja optimoivat ainakin GDD-prosessi, Carrollin skenaariomenetelmä sekä Cooperin prosessi.)

Sari A. Laakso (2006)

Suunnitteluprosessin valinta

Käyttöliittymäratkaisun kiinnitys

- Käyttöliittymäratkaisun kiinnittäminen eri menetelmissä:
 - Carrollin skenaariomenetelmässä ja Cooperin prosessissa käyttöliittymän tietosisältöä ja toimintalogiikkaa kiinnitetään tekstimuodossa skenaarioiden avulla.
 - Lauesenin Virtual Windowsissa ja GDD-prosessissa tietosisältö ja toimintalogiikka kiinnitetään piirtämällä näyttöjä.
 - Virtual Windowsissa kiinnitetään ensiksi tietosisältö, lopuksi toiminnot.
 - GDD-prosessissa kiinnitetään sekä tietosisältö että toiminnot yhtä aikaa.
- Käyttöskenaarioiden ja näyttökuvien etuna on se, että järjestelmän vaatimuksia voidaan arvioida/testata jo varhain.
 - Skenaarioiden avulla käyttäjät saavat paremmin selville todellisia tarpeitaan, koska he voivat simuloida järjestelmän toimintaa mielessään [Go04]. Suunnittelijat voivat arvioida skenaarioiden avulla suunnitteluratkaisuja ja paikantaa ongelmakohtia [Carroll94].
 - Näyttökuvien avulla järjestelmän toimintaa ja tietosisältöä voidaan testata varhaisessa vaiheessa (esim. simulointitestaus tai käytettävyytestaus).

Sari A. Laakso (2006)

GUIDe-malli ja GDD-suunnitteluprosessi

GUIDe-malliin sisältyvä simulointipohjainen GDD (*Goal-Derived Design*) on kehitteillä oleva, joissain yrityksissä käyttöön sovellettu käyttöliittymän suunnitteluprosessi (lisätietoja: Sari A. Laakso, salaakso@cs.helsinki.fi).

GDD:ssa kantavana ajatuksena on käyttää suunnittelussa samaa menettelyä kuin testauksessa ja järjestelmän todellisessa käytössä: käyttöliittymä suunnitellaan käyttötilanneasetelmien pohjalta.

Tällä kurssilla opetellaan GDD-prosessista yksinkertaistettua versiota, jossa mm. tavoitepohjaiset käyttötapaukset on korvattu yksinkertaisemmilla käyttötilannekuvauksilla.

GUIDen ja GDD:n periaatteet Vaiheet ja idea

- GUIDe-malli (Goals – User Interface Design – Implementation) [Laakso04] määrittelee rungon vaatimusten kartoittamiselle ja käyttöliittymän laatimiselle. Se rakentuu kolmesta karkeasta vaiheesta:
 1. käyttötilanteiden kerääminen vaatimuksiksi,
 2. käyttöliittymäratkaisun generointi simulointipohjaisella menetelmällä (Goal-Derived Design, GDD) sekä
 3. käyttöliittymän ja sen vaatiman toiminnallisuuden ohjelmoiminen em. vaiheiden *jälkeen*. Järjestelmää voidaan alkaa toteuttaa mitä tahansa prosessia noudattaen, esim. XP:n tai Scrumin pohjalta.
- GUIDe ja GDD perustuvat käyttötilannelähtöiseen suunnitteluun.

Sari A. Laakso (2015)

GUIDen ja GDD:n periaatteet

GUIDen päävaiheet

- Käyttötilanteiden kerääminen
 - Aluksi selvitetään kenttätutkimusten avulla käyttötilanteet, joita on tarkoitus tukea järjestelmän avulla. Nykykäytännöistä erotellaan niiden takana olevat tavoitteet ja ongelman virittämä ristiriita.
 - Käyttötilanteet kategorisoidaan siten, että saman datan organisoinnin virittämät tilanteet niputetaan yhteen. Suunnitteluun valitaan näistä edustavat instanssit ja tarvittavat variaatiot.
- Käyttöliittymäratkaisun generointi simuloimalla
 - Simuloitavaksi valitaan kerrallaan yksi käyttötilanne, jota ryhdytään piirtämään auki käyttöliittymäksi (tietosisältö ja toiminnot). Piirtämisen aikana kiinnitetään käyttötilanteelle mahdollisimman suoraviivainen minimaalinen toteuma: sekä käyttöliittymäratkaisu että työnkulku. Päätöksentekodata näytetään kerralla, ja käyttäjältä vaaditut säätämistoimet tämän näkymän esille saamiseksi minimoidaan. Tehokkuus optimoidaan.
 - Kun muutaman käyttötilanteen suorittaminen ja niissä tarvittu toimintalogiikka ja tietosisältö on piirretty samaan käyttöliittymään, viriää mahdollisesti refaktorointitarve. Refaktorointi tähtää yksinkertaisempaan ratkaisuun, joka kuitenkin virittyy samoista käyttötilanteista.

Sari A. Laakso (2015)

GUIDen ja GDD:n periaatteet

Vaatimusten selvittäminen

- GUIDen kahden ensimmäisen vaiheen sisällä on samoja piirteitä kuin mm.
- *ketterässä ohjelmistokehityksessä*: valitaan käyttötilanteita ja piirretään niitä vastaavaa käyttöliittymäratkaisua yksi tilanne kerrallaan, vrt. esim. käyttäjätarinat.
 - *testivetoisessa kehityksessä* (Test-Driven Development, TDD): suunnittelu pohjautuu samoihin käyttötilanteisiin, jotka ovat myös testitapauksia; tosin GUIDessa ne kategorisoidaan tietyllä tavalla ennen suunnittelun aloittamista.
 - *Lean-ohjelmistokehityksessä*: sekä käyttöliittymäratkaisussa että suunnitteluprosessissa tavoitellaan minimaalista ratkaisua, josta kaikki roska (waste) on eliminoitu.

Sari A. Laakso (2015)

GUIDen ja GDD:n periaatteet Asiakkaan rooli

- Yleensä asiakas osaa antaa tietoa saatujen käyttötilanteiden selvittämistä, priorisointia ja järjestelmän rajaamista varten.
 - Asiakkaan edustajalta saadaan tietoa keskeisistä kohderyhmistä, joilta käyttötilanteita kartoitetaan, ja hän pystyy hankkimaan kontakteja näiden edustajiin. Hänellä saattaa olla tietoa siitä, miksi tiettyä käyttötilannetta tulisi tukea tai miksi ei, tai tuetaanko ko. tilannetta nyt vai myöhemmin.
 - Asiakasta autetaan tämän tietämyksen käyttöön ottamiseksi, jotta hän voisi tehdä oman liiketoimintansa kannalta mahdollisimman hyviä päätöksiä.
- Sen sijaan GUIDessa oletetaan, että tyypillisesti asiakas ei voi osata
 - selvittää ja kategorisoida käyttötilanteita (selvitämme nämä hänelle) eikä
 - laatia siirtymää valituista käyttötilanteista mahdollisimman optimaaliseen, niitä tukevaan käyttöliittymäratkaisuun (teemme sen hänelle).

Tällä kurssilla opetellaan erityisesti näkemään tätä yhteyttä käyttötilanteiden ja käliratkaisun välillä, koska se on kriittinen taito, jonka hankkiminen vaatii vaivannäköä ja harjoittelua. Myös käyttötilanteiden selvittäminen helpottuu heti, kun tämä suunnittelutaito kehittyy.

Sari A. Laakso (2015)

GUIDen ja GDD:n periaatteet Muuttuvat vaatimukset/käli

Ketterässä kehityksessä (ilman GUIDea):

- Vaatimukset ovat asiakkaan vastuulla.
- Muuttuvia vaatimuksia pidetään normaalina ilmiönä, ja ne otetaan vastaan osana ohjelmiston kehitysprosessia.
- Asiakkaalta saadut muutokset tehdään toimivaan ohjelmaan.

GUIDessa:

- Vaatimusten selvittäminen on käyttöliittymätiimin vastuulla, ja asiakas toimii avustavan asiantuntijan roolissa.
- Vaatimusten muuttuminen nähdään pääosin illuusiona, joka syntyy siitä, ettei kukaan pyri selvittämään (tai osaa selvittää) niitä systemaattisesti, jolloin ne näyttävät muuttuvan.
- Käyttöliittymä muuttuu käyttötilanteiden pohjalta kälin suunnitteluvaiheessa, eikä mitään vielä ohjelmoida.

Sari A. Laakso (2015)

GUIDen ja GDD:n periaatteet Muuttuvat vaatimukset/käli

- GUIDessa muutoksia vaatimuksiin tulee,
 - jos jonkin osapuolen jokin keskeinen käyttötilanne on epähuomiossa jäänyt selvittämättä (=selvitystyössä tapahtunut virhe/puute), tai
 - jos asiakkaan liiketoimintamalli muuttuu (esim. lounasravintola päättääkin alkaa toimittaa ruokia myös kotiinkuljetuksella, tai lastentarvikekauppa ryhtyykin vuokraamaan hoitovälineitä myymisen sijaan).
- GUIDessa muuttuvien vaatimusten määrä minimoidaan.
 - Jäljelle jäävät vain ne muutokset, jotka johtuvat inhimillisen virheen korjaustarpeesta tai joita olisi ollut mahdotonta saada selville etukäteen.
 - Muutoksia ei synny siitä, että asiakas ei alussa tiedä riittävän täsmällisesti, mitä hän haluaa, tai siitä, että hän tietää paremmin vasta sitten, kun näkee, mihin hänen vaatimuksensa on johtanut. GUIDessakin tätä pidetään "normaalina" lähtötilanteena, mutta se ratkaistaan selvitystyöllä.
- GDD-prosessilla tuotettu käyttöliittymäratkaisu voi muuttua,
 - jos joku onnistuu löytämään vielä suoraviivaisemman ratkaisun kuin se, jonka suunnittelija osasi optimoida käyttötilanteiden pohjalta (=käyttöliittymäsuunnittelijan inhimillinen virhe/puute).

Sari A. Laakso (2015)

Käyttöliittymän piirtäminen GDD:lla

Suunnitteluun otetaan yksi käyttötilanne kerrallaan.

Vain käsillä olevaan tilanteeseen liittyvää tietosisältöä ja toimintoja piirretään näkyviin tehokkuutta optimoiden. Mieleen tulevat muut toiminnot, ideat, uhkat ja huolet kirjataan muistiin ongelmalistalle ja sivuutetaan.

Näyttökoko oletetaan rajattoman suureksi.

Usein kynällä paperille piirtäminen on nopein tapa tässä vaiheessa: suunnittelijan kognitiiviset resurssit kohdistuvat ensisijaisesti vaikeimpaan älylliseen työhön.

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla Suunnitteluperiaatteet 1-7

Käytä simuloinnin aikana seuraavia suunnitteluperiaatteita:

1. Vain tässä käyttötilanteessa tarvittavat tiedot ja toiminnot piirretään käyttöliittymään, ei mitään muuta. Loput ongelmallistalle.
2. Kaikki tiedot ja käyttöliittymän osat näytetään kerralla – sen sijaan, että järjestelmä pihtaa niitä ja käyttäjä pyytelee.
3. Käyttäjä pysyy yhdessä näkymässä koko ajan – sen sijaan, että hän kuljeksii järjestelmässä ympäriinsä näytöltä toiselle.
4. Käyttäjä säätää näkymää ja järjestelmä reagoi "lennossa".
5. Editointi paikallaan: Käyttäjä editoi tökkäsemällä suoraan sinne, missä kohde näkyy.
6. Päätöksentekoa simuloitaessa käyttäjän tulee osua vähintään kolmeen relevanttiin vaihtoehtoon: käyttäjä punnitsee näitä keskenään ja arvioi, mikä olisi paras tässä tilanteessa. Käyttäjälle tulee eteen myös huonoja vaihtoehtoja, jotka hän jättää sivuun.
7. [Valmiit ratkaisumallit (design patterns). Vähitellen suunnittelija alkaa tunnistaa simuloinnissa toistuvia, hyviä ratkaisumalleja.]

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla Suunnitteluperiaatteiden toiminta

- Suunnitteluperiaate 1 varmistaa, ettei järjestelmään synny ylimääräisiä tietoja tai toimintoja, joita ei tarvita missään käyttötilanteessa. Lisäksi sillä pyritään varmistamaan, että tarvittavat toiminnot ja tietosisältö on suunniteltu aina niitä hyödyntävien käyttötilanteiden näkökulmasta eikä yleisesti ripottelemalla ympäri järjestelmää (jolloin alkaa syntyä pitkiä vaivalloisia polkuja ja tehokkuus heikentyy).
- Suunnitteluperiaatteilla 2-5 pyritään optimoimaan mekaanista tehokkuutta jättämällä kaikki tarpeettomat suoritusvaiheet pois.
- Periaatteella 6 optimoidaan päätöksentekokohtien kognitiivista tehokkuutta (joskin samalla saadaan optimoitua myös näiden kohtien mekaaninen tehokkuus). Käyttäjältä pyritään poistamaan tarpeeton, tietojärjestelmän käytöstä syntyvä kognitiivinen työ, joka kuormittaa ja häiritsee varsinaista päätöksentekoa. Tarkoitus on vapauttaa mahdollisimman paljon kognitiivisia resursseja itse päätöksen prosessointiin.

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla Simulointi lyhyesti (tälläkin pärjää)

- Ota käyttötilanne 1 ja piirrä sen suorittaminen mahdollisimman suoraviivaisesti paperille siten, että noudatat annettuja suunnitteluperiaatteita, jotka auttavat optimoimaan tehokkuutta. Lopuksi simuloi tilanne alusta loppuun vielä pari kertaa yksityiskohtaisesti ja yritä poistaa turhia toimenpiteitä tai kognitiivista työtä, jos sellaisia ongelmia on ratkaisuun jäänyt.
- Ota käyttötilanne 2 ja yritä simuloida sen suorittaminen olemassa olevalla käyttöliittymällä samoin kuin edellä. Tee tarvittavat muutokset ja simuloi pariin kertaan. Lopuksi simuloi käyttötilanne 1, jotta voit korjata, jos se meni rikki.
- Ota käyttötilanne 3 ja yritä simuloida se edellisten sekaan. Tee tarvittavat muutokset. Simuloi edelliset uudelleen.
- Ota käyttötilanne 4 jne.

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla Simulointi päätöksentekolähtöisesti

- Tässä kuvataan 3-vaiheinen simulointitapa, jossa käyttötilanteen päätöksentekonäkymä rakennetaan aina ensin. Käytä annettuja suunnitteluperiaatteita.
- Vaiheet:
 1. Aloita piirtämällä käyttötilanteen keskeltä pelkkä päätöksentekonäkymä.
 2. Piirrä puuttuva alkupää: Piirrä käyttäjältä vaadittavat toimenpiteet järjestelmän alkutilasta em. päätöksentekonäkymään.
 3. Piirrä puuttuva loppupää: Piirrä käyttäjältä vaadittavat toimenpiteet päätöksentekonäkymästä käyttötilanteen suorittamisen loppuun asti.

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla

Vaihe 1: Päätöksentekonäkymä

- Piirrä käyttötilanteen 1 päätöksentekonäkymä näkyviin vertailussa tarvittavan tietosisällön osalta: mitä tietoa kohteista näytetään ja miten, jotta käyttäjä voisi vertailla relevantit vaihtoehdot ja päätyä parhaaseen lopputulokseen.
 - Kun valitset tietosisältöä ja sijoitat sitä näytölle, simuloi jatkuvasti käyttäjän toimintaa: mitä sellaista tietoa hän lukee, mikä vaikuttaa hänen päätöksentekoonsa. Sijoita näytölle ainoastaan tällaista tietosisältöä (sitä, mikä merkittiin simulointitestauksessa vertailumerkinnöillä).
 - Kiinnitä erityistä huomiota suunnitteluperiaatteeseen 6: Päätöksentekoa simuloitaessa käyttäjän tulee osua vähintään kolmeen relevanttiin vaihtoehtoon: käyttäjä punnitsee näitä keskenään ja arvioi, mikä olisi paras tässä tilanteessa. Käyttäjälle tulee eteen myös huonoja vaihtoehtoja, jotka hän jättää sivuun.
 - Käytä realistista dataa, jota tulee olla realistisen suuri määrä.
- Lähtökohtaisesti mitään toiminnallisuutta ei tarvita. Vasta kun jonkin toiminnon tuottama mekaaninen työ on ainoa keino vähentää kognitiivista työtä, lisää ko. toiminto (esim. vertailukori hyvien vaihtoehtojen poimimiseksi, mahdollisuus merkitä hylättyjä vaihtoehtoja visuaalisesti, ...).

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla

Vaihe 2: Alkutilasta päätöksentekoon

- Kun vaiheen 1 päätöksentekonäkymä on valmis, piirrä järjestelmän alkutila erilliselle paperille (kyseessä on sama näyttö, eri tila).
 - Koska teit vaiheessa 1 ensimmäisen käyttötilanteen päätöksentekonäkymän, tutki ensin, voisiko ko. näkymä olla suoraan järjestelmän alkutila, jolloin ei tarvittaisi mitään toiminnallisuutta tämän säätämiseksi näkyviin. Huomaa kuitenkin, että järjestelmä ei voi "arvata", mitkä kohteet juuri tämä käyttäjä tässä käyttötilanteessa tarvitsisi (esim. mitkä kirjaston kirjat tai mitkä autokaupan autot). Järjestelmä ei saa toimia maagisesti tai epädeterministisesti.
 - Näytä alkutilassa maksimaalisen paljon sellaista dataa, josta käyttäjälle on hyötyä juuri tässä tilanteessa.
- Laadi minimaalinen joukko toimenpiteitä, joiden avulla käyttäjä pystyy säätämään alkutilasta esille vaiheen 1 päätöksentekonäkymän.
 - Pidä käyttäjä koko ajan "samassa paikassa" ja anna hänen vain säätää näkymää (suorakäsittely, vrt. Direct Manipulation, esim. [Shneiderman98]).
 - Vältä navigointitarvetta viimeiseen asti.

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla

Vaihe 3: Päätöksenteosta loppuun

- Piirrä minimaalinen joukko toimenpiteitä, jotka tarvitaan käyttötilanteen loppuun saattamiseksi päätöksenteon jälkeen (esim. tilauksen lähettäminen, tuki paikan päälle pääsemiseksi).
- Simuloi käyttötilanteen suorittamisen loppuhäntä (esim. miten käyttäjä osaa kävellä bussipysäkiltä perille lomahotelliinsa tai messupaikalle).

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla

Seuraavat käyttötilanteet

Kun valitset variaatioita samasta tilanteesta:

- Kokeile ensin, onnistuuko variaation simulointi suoraan edellisten tilanteiden tuottamalla käyttöliittymällä. Jos ei, laadi tarvittavat minimaaliset muutokset tietosisältöön ja toiminnallisuuteen.

Kun valitset aidosti eri kategorian käyttötilanteen:

- Kun mukaan tulee eri kategorian käyttötilanne, siitä seuraa erilainen datan organisointi eli selvästi toisenlainen päätöksentekonäkymä kuin aiemmin piirtämäsi näkymät.
 - Piirrä ensin näkyviin uusi datanäkymä vastaavalla tavalla simuloimalla kuin "Kt 1:n päätöksentekonäkymä". Kokeile sitten simuloimalla, onko nämä mahdollista yhdistää yhdeksi näkymäksi niin, ettei kummankaan päätöksenteko häiriinny. Jos ei, säilytä uusi näkymä erillisenä. Voit sijoittaa sen esimerkiksi omalle välilehdelle (tab) aiemman rinnalle.
 - Piirrä käyttötilanne loppuun vastaavasti kuin "Kt 1:n piirtäminen loppuun". Alkutilana on nyt se alkutila, jonka olet jo aiemmin laatinut. Nyt on lisättävä minimaalinen joukko säätämistoimenpiteitä, joilla käyttäjä päätyy uuteen datanäkymään. Parhaassa tapauksessa aiemmin laaditut säätimet toimivat tässäkin.

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla Kohti realistista näyttökoko

- Kun olet piirtänyt tuen kaikille valitsemillesi käyttötilanteille ja refaktoroinut ratkaisua tarvittaessa, lopputuloksena on suuri näyttö, joka on jäsennetty ehkä parille välilehdelle. Miten tämä saadaan sovitettua tietokoneen näytölle tai jopa pieneen puhelimeen?
- Simuloi jokaisen kategorian keskeisiä käyttötilanteita yksi kerrallaan ja paikanna päätöksentekokohtien kriittiset vaiheet, joissa käyttäjä pitää mielessään vertailtavia tietoja tai työstää vaihteittain etenevää prosessia kohti ratkaisua.
- Säilytä kerralla näkyvissä ne tiedot, joita käyttäjä hyödyntää vertailussa (muussa tapauksessa hän joutuisi pitämään niitä mielessään tai navigoimaan edestakaisin näyttöjen välillä). Tee näyttöjaot kognitiivisesti kuormittavien kohtien väliin.
- Jos on tarpeen sovittaa todella pienelle näytölle, joudut katkaisemaan myös "huonoista kohdista". Yritä tällöin pitää samalla näytöllä ne tiedot, joiden pirstaloiminen kasvattaa mielessä pitämisen kuormaa eniten. Selvitä simuloimalla, missä ne ovat.

Sari A. Laakso (2015)

GDD-suunnitteluprosessi

Yhteenveto: Simuloiminen, testaus ja refaktorointi

Toimenpide a. Piirrä yksi käyttötilanne kerrallaan dataa ja toimintoja näkyviin 'yhteen suureen ikkunaan'.

Käyttötilanne 1

- 1.1 Piirrä kt 1 käyttöliittymäratkaisuksi simuloimalla käyttösekvenssiä vaihe vaiheelta.
- 1.2 Simuloi ilmeisimmät variaatiot.

Käyttötilanne 2

- 2.1 Editoi kt 2 edelliseen käyttöliittymään mukaan simuloimalla sitä vaihe vaiheelta.
- 2.2 Simuloi ilmeisimmät variaatiot.
- 2.3 Testaa käyttöliittymää simuloimalla edellinen käyttötilanne 1.

Käyttötilanne 3

- 3.1 Editoi kt 3 mukaan simuloimalla.
- 3.2 Simuloi ilmeisimmät variaatiot.
- 3.3 Testaa käyttöliittymää simuloimalla edelliset käyttötilanteet 1 ja 2.

Käyttötilanne 4

...

Toimenpide b. Testaa käyttötilanneketjut (vain mielekkäät tositalanteet).

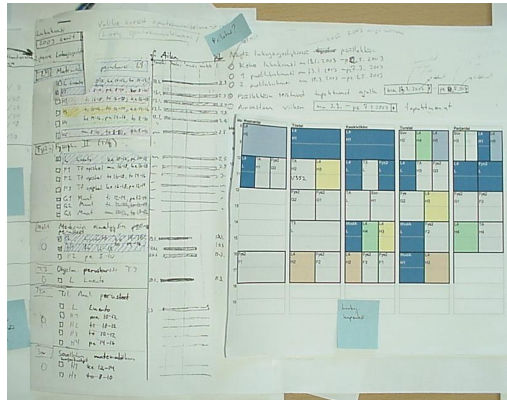
Simuloi esim. ketju 2, 1, 2, 4, 2, 2, 4.

Simuloi esim. ketju 1, 1, 3.

Toimenpide c. Refaktoroi käyttöliittymää tarvittaessa.

Sari A. Laakso (2006)

Kälin piirtäminen GDD:lla Esimerkki paperiprotosta



Simuloinnin aikana näyttökuvia kannattaa piirtää nopeasti kynällä paperille. Varo katkaisemasta simulointiprosessia turhaan.

Ne osat, joita on aidosti nopeampaa tehdä esimerkiksi PowerPointilla kuin käsin piirtämällä, voidaan tehdä tietokoneella ja tulostaa piirrosten joukkoon.

Esimerkki paperille laaditun näyttökuvan yhdestä sivusta:
WWWTopi (kurssi-ilmot ja opintojen suunnittelu)

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla Ongelmalistan käyttö

- Pysyttele tiukasti vain yhden käyttötilanteen piirtämisessä kerrallaan. Kerää ongelmalistaa, jos kesken piirtämisen mieleesi tulee...
 - uhka tai huoli, esim. mitäpä jos joku väärinkäyttää järjestelmää tai jos käyttäjä vahingossa tuhoaa jotain tai jos käyttäjä ei keksi, miten tätä käytetään (opittavuusuhka), tai jos käyttäjä pelästyy tms. käyttäjän tai asiakkaan hypoteettinen tunnetila,
 - jokin muu käyttötilanne, joka pitäisi huomioida,
 - toiminto, jota saatetaan tarvita muissa tilanteissa, tai
 - muissa tilanteissa ongelmalliseksi osoittautuvia ratkaisuja.
- Jos suunnittelija kesken piirtämisen huolestuu uhkatilanteista tai ryhtyy miettimään muissa käyttötilanteissa tarvittavia toimintoja, piirtäminen etenee hitaasti ja käyttöliittymästä tulee helposti sellainen, ettei sen avulla olekaan suoraviivaista hoitaa yhtään käyttötilannetta.

Sari A. Laakso, Antti Latva-Koivisto (2006)

Kälin piirtäminen GDD:lla Ongelmalistan purkaminen 1/2

- Kun simulointi etenee, ongelmalistalle kerätyt kohteet yleensä alkavat vähentyä, koska uusien käyttötilanteiden mukaanotto ratkaisee aiempia ongelmia "itsestään".
- Kun olet piirtänyt kaikki käyttötilanteet mukaan ratkaisuun, käy ongelmalista läpi. Vedä yli ne ongelmat, jotka tuli jo hoidettua uusien tilanteiden simuloinnin yhteydessä.
- Käy jäljelle jääneet ongelmat läpi. Jos listalla on...
 - ...uhka tai huoli: Selvitä, mihin käyttötilanteeseen se liittyy ja onko uhka todellinen vai kuviteltu. Ota tarvittaessa yhteyttä asiakkaaseen (kysyäksesi esim. liiketoiminnan tavoitteista) tai käyttäjiin (kysyäksesi esim. käyttötilanteista tai nykykäytännöistä) tai muihin asiantuntijoihin. Jos uhka on todellinen, ota se huomioon ja korjaa ratkaisua. Selvitä, onko se mahdollista hoitaa esim. tarjoamalla perumismahdollisuus tai parantamalla opittavuutta.

Sari A. Laakso (2015)

Kälin piirtäminen GDD:lla Ongelmalistan purkaminen 2/2

- ...uusi toiminto, jota arvelet jonkun käyttäjän tarvitsevan: Etsi yksi konkreettinen käyttötilanne, jossa kyseistä toimintoa voitaisiin tarvita. Jos sellaista ei löydy, jätä toiminto sivuun. Jos löytyy, katso nyt hetken ajan pelkästään löytämäsi käyttötilannetta ja piirrä sen tukemiseksi mahdollisimman suoraviivainen ratkaisu. Paras ratkaisu ei välttämättä olekaan alunperin mieleesi tullut toiminto.
- ...uusia käyttötilanteita: Kokeile ensin simuloimalla, onnistuuko niidenkin tekeminen. Jos ei onnistu, mutta tilannetta olisi syytä tukea, ota uusi käyttötilanne suunnitteluun mukaan.
- ...käyttöliittymän aiheuttama ongelma jonkin muun kuin suunnittelussa mukana olevan käyttötilanteen kannalta: Selvitä ensiksi, mikä tämä muu tilanne on. Sen jälkeen arvioi, onko haitta todellinen. Jos sen korjaaminen heikentää muiden käyttötilanteiden suorittamista, haarukoi sopiva kompromissiratkaisu simuloimalla kaikkia ongelmaan liittyviä käyttötilanteita nopeasti peräkkäin.

Antti Latva-Koivisto, Sari A. Laakso (2006)

GDD-suunnitteluprosessi

Esimerkki 1: PDA-bussiaikataulut



Kuva:
www.infosatellite.com/news/2002/03/a140302mmo2_xda.html

Sari A. Laakso (2003)

Tässä esimerkissä demotaan GDD-suunnitteluprosessin etenemistä.

Suunnittelun kohdelaitteena on PDA:

- Syöttölaitteena on kynä.
- Laitteessa on GPS-paikannus, joka tietää laitteen sijainnin noin viiden metrin tarkkuudella.

Tässä käyttöliittymä on luentomateriaalin tiivistämisen vuoksi piirretty pienelle näytölle, vaikka todellisuudessa suunnittelu tehtäisiin ensin ilman näyttörajoitteita. Esimerkistä kuitenkin näkyy selvästi, miten tietosisältö ja erityisesti toiminnot kertyvät käyttöliittymään simuloinnin edetessä.

Käyttötilanteet

Kt 1 ja 2



Kt 1: Tältä pysäkillä vanhempien luo Paloheinään

Joonaksen tavoite: Joonas on Töölön tulin bussipysäkillä klo 15.55. Hän on menossa pitkää aikaa tapaamaan Paloheinässä asuvia vanhempiaan.

Tilatietoja:

- Ma 7.7. klo 15.55 Joonas on Töölön tulin bussipysäkillä (Mannerheimintie 71)
- Hän on juuri käynyt ostoksilla
- Hänen vanhempansa ovat asuneet viimeiset 20 vuotta Paloheinässä osoitteessa Raivontie 2
- Hän tietää, että bussilla 63 pääsee Mannerheimintien pysäkeiltä vanhempien luo
- Hän tietää, että busseja 63 menee noin kymmenen minuutin välein

Sari A. Laakso (2003)

Kt 2: Tältä pysäkillä eläinlääkäriin Kannelmäkeen

Hannan tavoite: Hanna on koiransa kanssa Töölön tulin bussipysäkillä klo 15.55. Eläinlääkärin vastaanotto on Kannelmässä klo 16.30.

Tilatietoja:

- Ma 7.7. klo 15.55 Hanna on koiransa kanssa Töölön tulin bussipysäkillä (Mannerheimintie 71)
- Hän on juuri käynyt ostoksilla
- Hän on varannut eläinlääkärin määräkoiralleen Kannelmäkeen (Soittajantie 1) klo 16.30:ksi
- Hän on käynyt siellä kerran aiemminkin tuttavien autokyydillä
- Hän tietää, että busseja menee Kannelmäkeen melko usein, muttei tiedä Kannelmäen bussien numeroita

Käyttötilanteet Kt 3

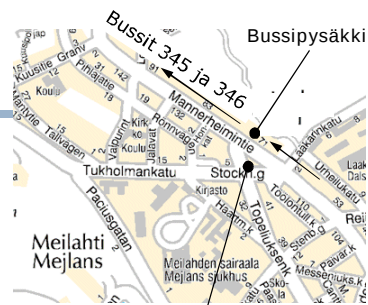
Kt 3: Kahvilasta isovanhempien luo Takkulaan

Jannen tavoite: Janne on Tullinpuomin Shellillä kahvilla klo 15.55. Hän on menossa iltapäivällä tervehtimään Takkulassa asuvia isovanhempiaan.

Tilatietoja:

- Ma 7.7. klo 15.55 Janne on Tullinpuomin Shellillä kahvilla
- Hänen isovanhempansa ovat asuneet viimeiset 15 vuotta Espoon Takkulassa (Takkulantie 6)
- Hän tietää, että busseilla 345 ja 346 pääsee Mannerheimintien pysäkeiltä isovanhempien kotiin
- Hän tietää, että bussi 345 jää kauemmas (noin 1 km:n kävely), mutta bussilla 346 pääsee aivan lähelle (muutama sata metriä)

Sari A. Laakso (2003)



Tullinpuomin Shellin kahvio



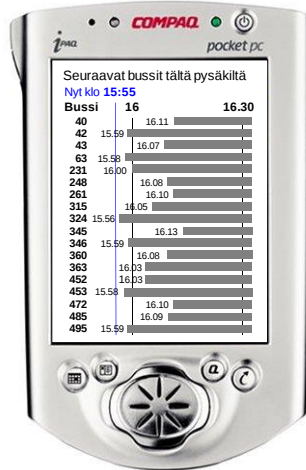
Kartta: pathfinder3.meridian.fi/ytv/fi/

Bussiaikatauluesimerkki GDD-demonon valitut käyttötilanteet

- Kt 1: Joonas vanhempien luo Paloheinään
 - Piirrä käyttöliittymäratkaisuksi simuloimalla vaihe vaiheelta.
 - Simuloi ilmeisimmät variaatiot.
- Kt 2: Hanna eläinlääkäriin Kannelmäkeen
 - Editoi kt 2 mukaan edelliseen käyttöliittymään simuloimalla.
 - Simuloi ilmeisimmät variaatiot.
 - Testaa simuloimalla edellinen käyttötilanne 1.
- Käyttötilanneketjut (ei välttämätöntä testata juuri tässä vaiheessa)
 - Testaa ketju 2->1.
- Kt 3: Janne isovanhempien luo Takkulaan
 - Editoi kt 3 mukaan edelliseen käyttöliittymään simuloimalla.
 - (Simuloi ilmeisimmät variaatiot. Jätetty tästä pois.)
 - Testaa simuloimalla edelliset käyttötilanteet 1 ja 2.

Sari A. Laakso (2003)

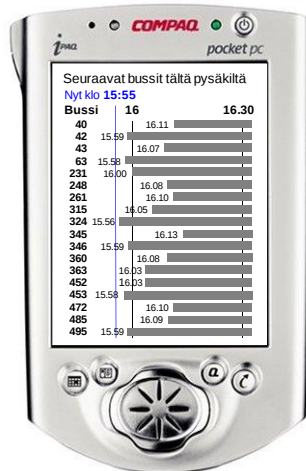
Kt 1: Joonas vanhempien luo Piirrä käyttöliittymäratkaisu simuloimalla



Tässä päätöksentekonäkymä sijoitetaan suoraan etusivulle, koska muita käyttötilanteita ei vielä ole suunnittelussa mukana. Yhtään toimintoa tai navigointiaskelta ei vielä tarvita.

Sari A. Laakso (2003)

Kt 1: Joonas vanhempien luo Simuloi ilmeisimmät variaatiot

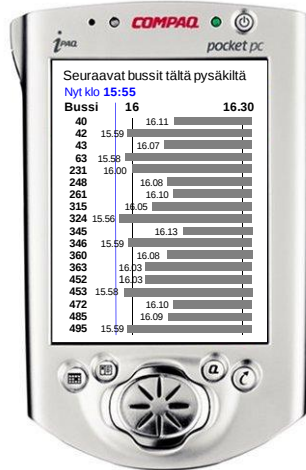


Variaatio 1
Joonaksen vanhemmat asuvat Vantaalla osoitteessa Emännänkuja 1. Joonas tietää, että busseilla 453, 472 ja 485 pääsee vanhempien kotitalolle.

Sari A. Laakso (2003)

Kt 2: Hanna eläinlääkäriin

Yritä tehdä kt 2 suoraan edellisellä kälillä



Sari A. Laakso (2003)

Ensin katsotaan, sattuisiko edellinen käyttöliittymä toimimaan suoraan tässäkin käyttötilanteessa 2.

Ongelma: Hanna ei selviytyisi Kannelmäkeen eläinlääkäriin Joonaksen käyttöliittymäratkaisulla ollenkaan!

Piirrämme uuden datanäkymän: mitä Hannan tilanteessa tarvitaan? Hänelle on näytettävä ne bussit, jotka menevät oikeaan paikkaan (selvitetään oikea vastaus: 42 ja 43). Eläinlääkäriin vastaanottoaika on 16.30, joten hyvissä ajoin sitä ennen on oltava perillä (saapumisaika).

Piirretään ensin seuraavan sivun kolmas näkymä, minkä jälkeen mahdollisimman vähillä säädöillä etusivulta reitti sinne (kaksi ensimmäistä näyttökuvaa).

Kt 2: Hanna eläinlääkäriin

Editoi kt 2 edelliseen káliin mukaan

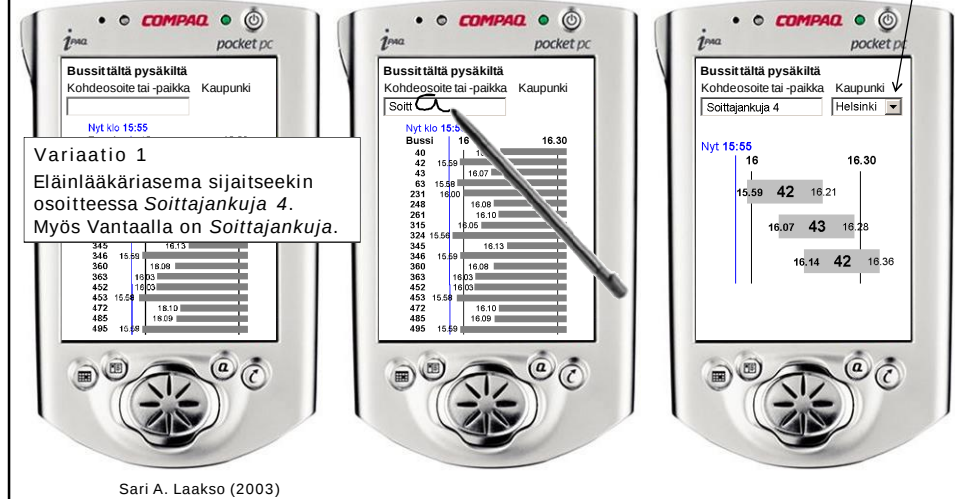


Sari A. Laakso (2003)

Kt 2: Hanna eläinlääkäriin

Simuloi ilmeisimmät variaatiot

Editoidaan pieni muutos.



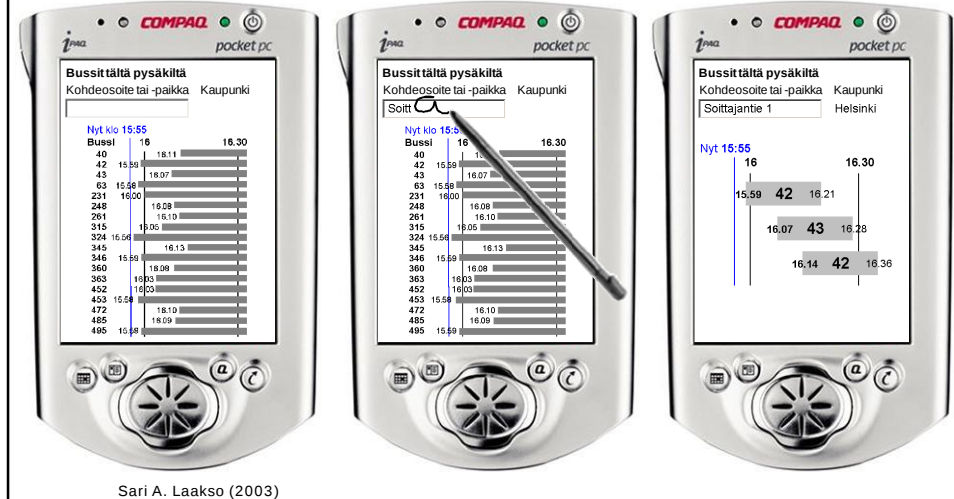
Testaa edelliset tilanteet

Testaa simuloimalla kt 1



Käyttötilanneketjut

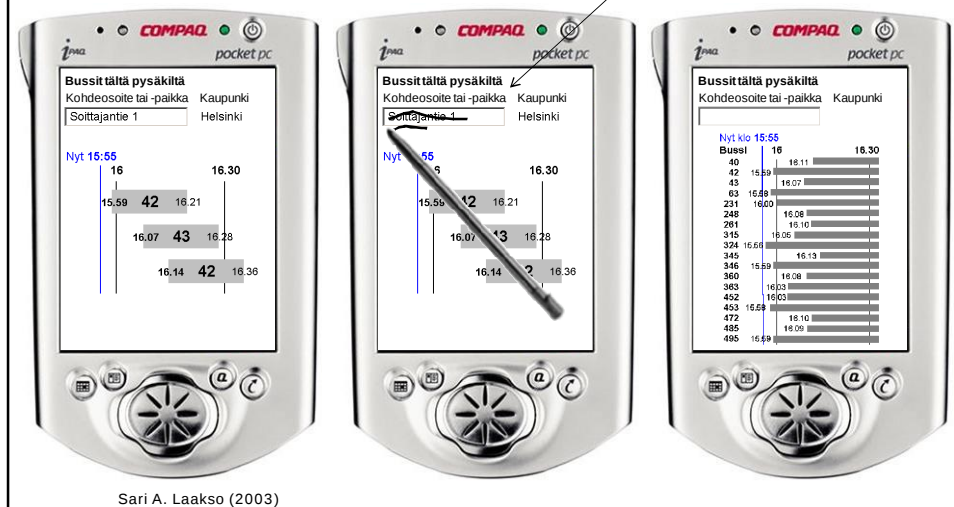
Testaa ketju: Kt 2 -> Kt 1



Käyttötilanneketjut

Testaa ketju: Kt 2 -> Kt 1

Ketjun testauksessa ilmeni tarve uudelle tyhjennä-toiminnolle. Lisätään se.

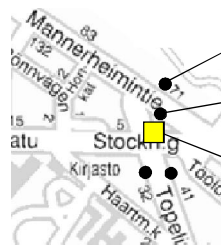


Kt 3: Janne Takkulaan

Yritä tehdä kt 3 suoraan edellisellä kälillä



Ongelma: Entinen käli ei toimi, koska Janne on nyt Shellin kahviossa, eikä lähin pysäkki ole se, jolta hänen tietämänsä bussit 345 ja 346 menevät Takkulaan. Laite näyttää väärää pysäkkiä ja pelkkiä ensimmäisiä busseja.



Takkulaan menevät bussit 345 ja 346

Lähin pysäkki, jonka busseja laite näyttää

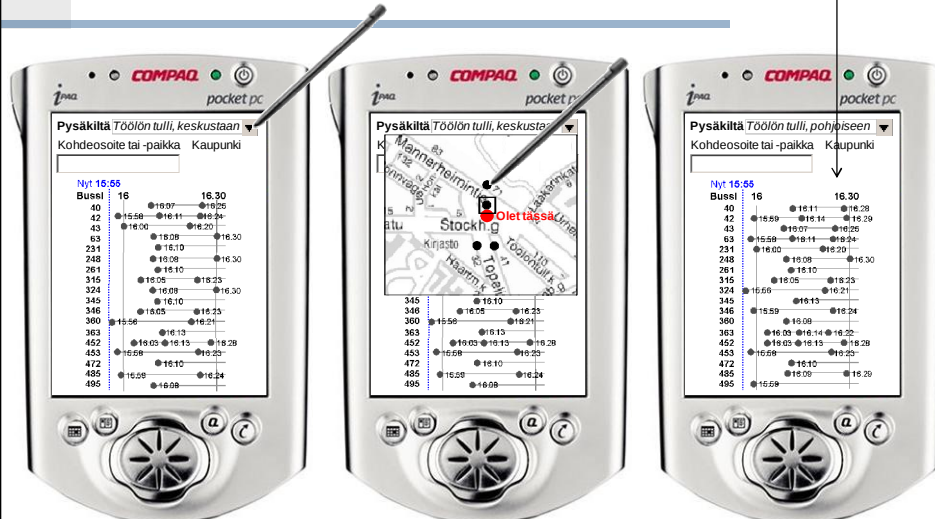
Shellin kahvio

Sari A. Laakso (2003)

Kt 3: Janne Takkulaan

Editoi kt 3 edelliseen käliin mukaan

Tehdään uusi datanäkymä. Se pystytään integroimaan entiseen.



Sari A. Laakso (2003)

Sari A. Laakso (2015)

Testaa edelliset tilanteet

Testaa simuloimalla kt 1



Sari A. Laakso (2003)

Testituloksena ongelma: Joonaksen pitäisi päästä lähtemään *tältä* pysäkillä, jolle hän on saapunut, mutta laite näyttää sen pysäkin busseja, jonka käyttäjä on valinnut *edellisellä* kerralla.

Suoraviivaisin tapa päästä takaisin kiinni *lähimpään* pysäkkiin? (Tarvitaan myös Hannan kt 2:ssa.)

Käyttäjä voisi valita Pysäkillä-valikosta *lähimmän* pysäkin käsin, joten tämä tilanne on mahdollista suorittaa nykyisenkin kälin avulla. Päättämme kuitenkin suoraviivaistaa *lähimmässä* pysäkissä kiinni pysymistä, ja lisäämme uuden toiminnon, koska emme keksi, miten voisimme suoraviivaistaa olemassa olevaa valikkovalintaa.

Testaa edelliset tilanteet

Korjaa kt 1:n käyttöliittymää



Lisätään mahdollisuus pysyä kiinni lähimmässä pysäkissä.



Sari A. Laakso (2003)

Testaa edelliset tilanteet

Testaa simuloimalla kt 1



Sari A. Laakso (2003)

Testitulos: Kt 1 toimii riittävän hyvin.

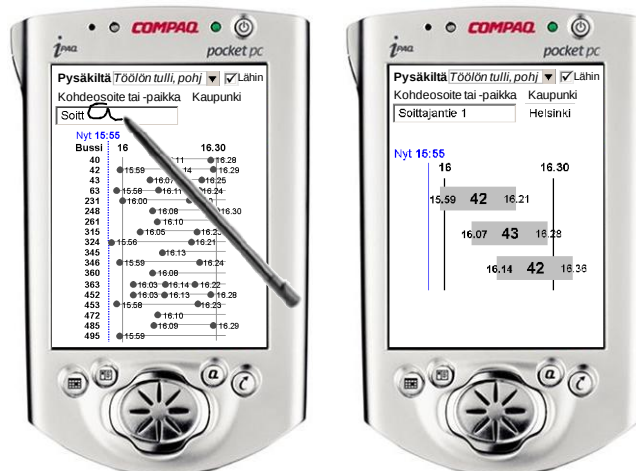
Huomio: Niin kauan kuin Joonaksella on *Lähin*-checkbox valittuna ja hakukenttä tyhjänä, hän pääsee edelleen tavoitteeseensa suorinta mahdollista tietä: vain vilkaisemalla laitetta tekemättä mitään.

Muissa tilanteissa (riippuen edeltävästä käyttötilanteesta) hän joutuu

- klikkaamaan *Lähin*-checkboxin päälle ja/tai
- tyhjentämään hakuehtokentän.

Testaa edelliset tilanteet

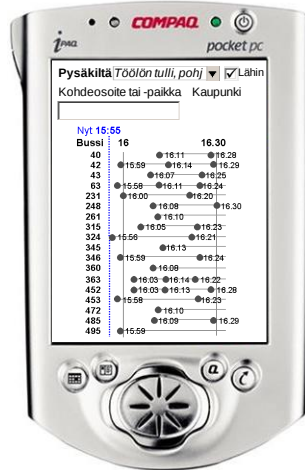
Testaa simuloimalla kt 2



Sari A. Laakso (2003)

Käyttötilanneketjut

Testaa ketju: Kt 3 -> Kt 1 -> Kt 2 -> ...

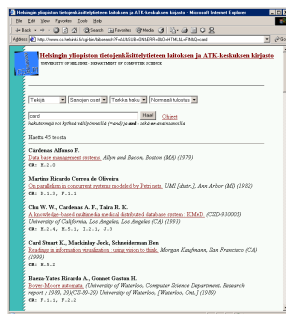


Testausta voitaisiin jatkaa uudella käyttötilanteiden ketjulla, minkä jälkeen käyttöliittymään integroitaisiin uusi käyttötilanne 4.

Sari A. Laakso (2003)

GDD-suunnitteluprosessi

Esimerkki 2: Kirjastohaku



TKTL:n vanha kirjastohakujärjestelmä

Tässä esimerkissä suunniteltavana käyttöliittymänä on yliopiston tietojenkäsittelytieteen laitoksen (TKTL) kirjastojärjestelmä.

Lähtökohtana on käyttötilanteita, joiden pohjalta käyttöliittymää piirretään GDD-suunnitteluprosessin mukaisesti tilanne kerrallaan.

Huom. Kaikkia simuloinnin välivaiheita ei ole piirretty näkyviin tässä esimerkissä. Esimerkin tarkoitus on havainnollistaa, miten tietosisältö ja toiminnallisuus kertyvät järjestelmään, kun uusia käyttötilanteita simuloidaan mukaan.

Kirjastoexamplesimerkki

Käyttötilanne 1

Kt 1: Cardin visualisointikirja omasta lähikirjastosta

Tutkijan tavoite: Hannu Toivonen tietää, että Cardin kirjassa olisi hyviä glyyfesimerkkejä hänen *Tutkimustiedonhallinnan peruskurssin* luentoaan varten, mutta hänellä ei ole Cardin kirjaa.

Tilatietoja:

- Tänään on ma 1.9. klo 9.30.
- Luento on ti 9.9. klo 10-12.

Cardin kirja: Information Visualization

- Hannu on aiemminkin lukenut Cardin kirjaa, mutta hänellä ei ole omaa kappaletta kirjasta.
- Omassa tietojenkäsittelytieteen laitoksen (TKTL) lähikirjastossa on 3 kpl, joista 1 on lainassa Inkeri Verkamolla ja 2 hyllyssä.

Sari A. Laakso (2003)

Käyttötilanne 1 (Cardin kirja lähikirjastosta)

Kirjan haku

Kansi	Tekijä(t)	Kirjan nimi	Vuosi	Hyllyssä
Hae	card			
	Card R. Ny, Dumas Eric, Mevel Franck	The Linux Kernel Book	1998	1
	Card Stuart, Mackinlay Jock, Shneiderman Ben	Readings in Information Visualization: Using Vision to Think	1999	2
	Card Stuart, Moran Thomas, Newell Allen	The Psychology of Human-Computer Interaction	1983	1
	Card elli Luca	Typeful Programming	1989	
	Card enas Alfonso	Research Foundations in Object-Oriented and Semantic Database Systems	1990	1
	Iyer Balakrishna, Ricard Gray, Varman Peter	An Efficient Percentile Partitioning Algorithm for Parallel Sorting	1989	2

Sari A. Laakso (2003)

Kirjastoexamplesimerkki

Käyttötilanne 2

Kt 2: Kaikki Cardin visualisointikirjat lainassa

Tutkijan tavoite: Hannu Toivonen tietää, että Cardin kirjassa olisi hyviä glyyphiesimerkkejä hänen *Tutkimustiedonhallinnan peruskurssin* luentoaan varten, mutta hänellä ei ole Cardin kirjaa.

Tilatietoja:

- Tänään on ma 1.9. klo 9.30.
- Luento on ti 9.9. klo 10-12.

Cardin kirja: Information Visualization

- Hannu on aiemminkin lukenut Cardin kirjaa, mutta hänellä ei ole omaa kappaletta kirjasta.
- Omassa tietojenkäsittelytieteen laitoksen (TKTL) lähikirjastossa on 3 kpl. jotka kaikki ovat lainassa:
 - Juha Tainalla,
 - Inkeri Verkamolla ja
 - Hannu Erkiöllä.

Sari A. Laakso (2003)

Käyttötilanne 2 (kaikki lainassa)

Kirjan haku

Kansi	Tekijä(t)	Kirjan nimi	Vuosi	Hyllyssä Lainassa		
				Pvm	Lainaja	Puhelin
Hae	card					
	Card Ray, Dumas Eric, Mevel Fränck	The Linux Kernel Book	1998	1		
	Card Stuart, Mackinlay Jock, Shneiderman Ben	Readings in Information Visualization: Using Vision to Think	1999		18.3.03 Juha Taina 31.3.03 Inkeri Verkamo 15.5.03 Hannu Erkiö	44 226 44 219 44 253
	Card Stuart, Moran Thomas, Newell Allen	The Psychology of Human-Computer Interaction	1983	1		
	Cardelli Luca	Typeful Programming	1989		10.6.03 Petri Myllymäki 12.7.03 Mikael Jokela	44 173 44 628
	Cardenas Alfonso	Research Foundations in Object-Oriented and Semantic Database Systems	1990	1		
	Iyer Balakrishna, Ricard Gray, Varman Peter	An Efficient Percentile Partitioning Algorithm for Parallel Sorting	1989	2		

Sari A. Laakso (2003)

Sari A. Laakso (2015)

Kirjastoexamplesimerkki

Käyttötilanne 3

Kt 3: Waren visualisointikirjaa ei kirjastossa

Tutkijan tavoite: Hannu Toivonen tietää, että Waren kirjassa olisi *Tietokonegrafiikan* kurssille hyviä esimerkkejä rinnakkais-koordinaattien käytöstä, mutta hänellä ei ole Waren kirjaa.

Tilatietoja:

- Tänään on ma 1.9. klo 9.30.
- Luento on ti 9.9. klo 10-12.

Waren kirja: Information Visualization

- Hannu on aiemminkin lukenut Waren kirjaa, mutta hänellä ei ole omaa kappaletta kirjasta.
- Omassa tietojenkäsittelytieteen laitoksen (TKTL) lähikirjastossa ei ole kirjasta lainakappaleita ollenkaan.
- Hannu ja hänen opiskelijansa tarvitsevat kirjaa todennäköisesti myöhemminkin, koska kirja on mm. hänen *Tietokonegrafiikka*-kurssinsa oheismateriaalina.

Sari A. Laakso (2003)

Käyttötilanne 3 (kirjaa ei ole kirjastossa ollenkaan)

Kirjan haku

Kansi Tekijä(t)		Kirjan nimi	Vuosi	Hyllyssä Lainassa		
				Lainapvm	Lainajan nimi	Puhelin
Hae	ware					
☐	Neeser, Andy, <u>Ware</u> Malcom	Design of a V.34 modem for a real-time multitasking DSP operating system	1995	1		
☐	Putnam Lawrence, Myers <u>Ware</u>	Measures for excellence : reliable software on time, within budget	1992	1		

Sari A. Laakso (2003)

Simulointipohjainen käyttöliittymäsuunnittelu

Kirjan haku

Hankintaehdotus

Kansi Tekijä(t)		Kirjan nimi	Vuosi	Hyllyssä	Lainassa	
				Lainapvm	Lainajan nimi	Puhelin
Hae	ware					
☐	Neeser Fredy, <u>Ware</u> Malcom	Design of a V.34 modem for a real-time multitasking DSP operating system	1995	1		
☐	Putnam Lawrence, Myers <u>Ware</u>	Measures for excellence : reliable software on time, within budget	1992	1		

Sari A. Laakso (2003)

Kirjan haku

Hankintaehdotus

Tee hankintaehdotus kirjastoon

Pvm	Tekijä(t)	Kirjan nimi
24.10.03	Ware Colin	Information visualization: perception for design

Lähetä kirjastoon

Tyhjennä

Lähetetyt hankintaehdotukset

Pvm	Tekijä(t)	Kirjan nimi

Käsitellyt hankintaehdotukset

Tila	Tekijä(t)	Kirjan nimi

Hankittava kirja

Kustantaja

Morgan Kaufmann

Painopaikka

San Diego, CA, USA

Painovuosi

2000

Painos

1.

ISBN

1-55860-511-8

Hinta-arvio

?

Tarvittavien lainakappaleiden lkm

2

Perustelut

Tietokonegrafiikka-kurssin oheismateriaalia. Lähdemateriaaliksi Visualisointi-seminaariin.

Sari A. Laakso (2003)

Kirjan haku

Hankintaehdotus

Tee hankintaehdotus kirjastoon

Pvm	Tekijä(t)	Kirjan nimi
24.10.03		

☐ Kirjan nimi
☐ Kirjailijan nimi

Lähetetyt hankintaehdotukset

Pvm	Tekijä(t)	Kirjan nimi
24.10.03	Ware Colin	Information visualization: perception for design

Käsitellyt hankintaehdotukset

Tila	Tekijä(t)	Kirjan nimi

Hankittava kirja

Kustantaja Morgan Kaufmann
 Painopaikka San Diego, CA, USA
 Painovuosi 2000
 Painos 1.
 ISBN 1-55860-511-8
 Hinta-arvio ?
 Tarvittavien lainakappaleiden lkm 2

Perustelut

Tietokonegrafiikka-kurssin oheismateriaalia.
 Lähdemateriaaliksi Visualisointi-seminaariin.

Sari A. Laakso (2003)

Kirjastoexamplesimerkki

Käyttötilanne irti järjestelmästä

- Älä kiinnitä käyttötilanteessa ratkaisutapoja. Esim. *"kirjan varaaminen"* ei koskaan ole käyttäjän tavoite, vaan kiinnitetty ratkaisutapa (tehtävä tai toiminto).
- Miksei kirjan varaamista pidä kiinnittää käyttötilanteessa?
 - Laaditaan ensin varauksen tekemisen tarpeeseen osuva kunnollinen käyttötilanneasetelma ja katsotaan, millainen käyttöliittymäratkaisu siitä seuraa: Otetaan pohjaksi aiempi käyttötilanne 2, mutta tehdään siitä uusi käyttötilanne 4 asettelemalla tarjonta niin, että omassa lähikirjastossa kaikki kirjat ovat varattuja, mutta kirjaa olisi saatavana muissa kirjastoissa.
 - Piirretään käyttötilannetta 4 optimoiva käyttöliittymäratkaisu. Se vie nyt päätöksentekokohdan kartalle (vahva sijaintiriippuvuus), jos oletamme työnkulun pysyvän ennallaan ("liiketoimintamalli" ei muutu).

Kirjasto-esimerkki

Käyttötilanne 4

Kt 4: Cardin visualisointikirja yliopiston muissa kirjastoissa

Tutkijan tavoite ja ongelma: Hannu Toivonen pitää *Tutkimus-tiedonhallinnan peruskurssin* luennon taas ensi viikolla, mutta hänellä ei ole sopivia glyyphiesimerkkejä vielä. Hän tietää, että Cardin kirjassa olisi hyviä esimerkkejä ja kuvia, mutta hänellä ei ole kirjaa.

Tilatietoja:

- Tänä on ma 1.9. klo 9.30. Luento on ti 9.9. klo 10-12.

Tarvittava kirja

- Hannu on aiemmin lukenut kirjaa, mutta hänellä ei ole omaa kappaletta.
- Hannu muistaa, että kirjan nimi on jokin 'Information Visualization' ja yksi tekijöistä on Card.
- Kirjan todelliset lähdetiedot: Card Stuart, MacKinlay Jock, Shneiderman Ben, *Readings in Information Visualization: Using Vision to Think*.

Kirjan saatavuus

- Omassa TKTL:n lähikirjastossa on 3 kpl, kaikki lainassa.
- Läketieteellisen kirjastossa on 3 kpl, lainassa 2, saatavilla 1.
 - Hannu ei ole aiemmin käynyt siellä, ei tiedä kirjaston sijaintia.
- Kasvatustieteellisen ja psykologian kirjastoissa 2 kpl, kaikki lainassa.

Copyright © 2006 / Sari A. Laakso

Ero kt2:een

Kirjan haku

Omat varaukseni

Hankintaehdotus

Kansi	Tekijä(t)	Kirjan nimi	Vuosi
Hae	card		
	Card Remy, Dumas Eric, Mevel Franck	The Linux Kernel Book	1998
	Card Stuart, Mackinlay Jock, Shneiderman Ben	Readings in Information Visualization: Using Vision to Think	1999
	Card Stuart, Moran Thomas, Newell Allen	The Psychology of Human-Computer Interaction	1983
	Cardelli Luca	Typeful Programming	1989
	Cardenas Alfonso	Research Foundations in Object-Oriented and Semantic Database Systems	1990
	Iyer Balakrishna, Ricard Gray, Varman Peter	An Efficient Percentile Partitioning Algorithm for Parallel Sorting	1989

Löytyi 26 / 48 559 kirjaa

Kirjan saatavuus
(Lainassa olevien lkm sulkeissa)

Tietojenkäsittelytiede (3)
Teollisuuskatu 23, ark. 9-16

- Esko Ukkonen p. 44 226 18.3.03
- Inkeri Verkamo p. 44 219 31.3.03
- Hannu Erkö p. 44 253 15.5.03

Läketieteellinen
Haartmaninkatu 3, ark. 10-16

Saatavilla: **1** Varaa

Psykologia (2 lainassa)
Siltavuorenpenger 20 A, ark. 9-15

Kasvatustieteellinen (2 lainassa)
Bulevardi 18, ark. 8-18

Simulointipohjainen käyttöliittymäsuunnittelu

Kirjan haku

Omat varaukseni

Hankintaehdotus

Voimassa olevat varaukset

Lääketieteellinen
Haartmaninkatu 3, ark. 10-16

Varatut kirjat
Card, Mackinlay, Shneiderman:
**Readings in Information
Visualization: Using Vision to
Think.**
Varattuna ma 10.11. – ke 12.11.

Peru varaus

Tietojenkäsittelytiede
Teollisuuskatu 23

Ratkaisun suoraviivaistaminen?
Suurin ylimääräinen työ syntyy nyt
kirjan noutamisesta.

Kirjan haku

Hankintaehdotus

Kansi	Tekijä(t)	Kirjan nimi	Vuosi	Lähikirjasto (TKTL)		Muut kirjastot	
				Hyllyssä	Lainassa	Hyllyssä	Toimitusaika (lainassa)
	card Remy, Dumas Eric, Mevel Franck	The Linux Kernel Book	1998	1			
	Card Stuart, Mackinlay Jock, Shneiderman Ben	Readings in Information Visualization: Using Vision to Think	1999		18.3.03 Esko Ukkonen 44 226 31.3.03 Inkeri Verkamo 44 219 15.5.03 Hannu Erkiö 44 253	1 (6)	Tänään ma 10.11. klo 15 mennessä
	Card Stuart, Moran Thomas, Newell Allen	The Psychology of Human-Computer Interaction	1983	1		(3)	
	Cardelli Luca	Typeful Programming	1989		10.6.03 Petri Myllymäki 44 173 12.7.03 Mikael Jokela 44 628		
	Cardenas Alfonso	Research Foundations in Object-Oriented and Semantic Database Systems	1990	1			

Löytyi 26 / 48 559 kirjaa

Tilaa sisäpostissa

Omat tilaukseni

Tekijä(t)	Kirjan nimi	Toimitus

Peru tilaus

Työnkulkumallit

Varaus-tehtävä on kiinnitetty ratkaisu

■ Työnkulkumalli 1

- Hannu noutaa vapaana olevan lainakappaleen kirjastosta.
- Hannu käy palauttamassa kirjan kirjastoon.

Varaus-tehtävä (task) on tässä tapauksessa mielekäs, jotta Hannu ei lähtisi kirjastoon turhaan.

■ Työnkulkumalli 2

- Hannu tilaa vapaana olevan lainakappaleen kirjastosta työpaikalleen. Kirja toimitetaan sisäpostissa perille vielä saman päivän aikana.
- Hannu palauttaa kirjan lähettämällä sen kirjan mukana tulevilla ohjeilla takaisin kirjastoon.

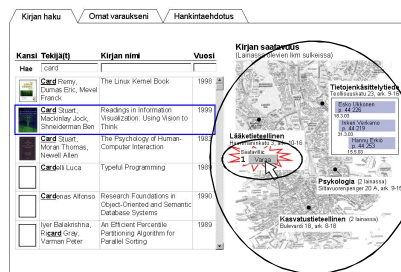
Tässä työnkulussa erillistä Varaus-tehtävää ei enää ole.

Copyright © 2006 / Sari A. Laakso

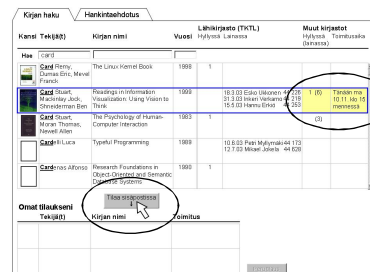
Työnkulkumallin vaihtaminen

Päätöksenteko muuttuu

- Noutamismenettelyssä Hannu tarvitsee kartalle suhteutettuja kirjastoja päätöksentekonsa tueksi:



- Sisäpostimenettelyssä on samantekevää, missä kirjasto sijaitsee:



Copyright © 2006 / Sari A. Laakso

GDD-suunnitteluprosessi

Tuloksena hyödyllisyys ja käytettävyys

- Käyttötilanteiden simuloinnin avulla piirretystä käyttöliittymästä nähdään
 - tarvittava tietosisältö ja
 - toiminnot.
- Konkreettisista käyttötilanteista saadut toiminnalliset vaatimukset on kuvattu
 - ymmärrettävässä ja
 - testauskelpoisessa muodossa.

Hyllyssä		Lainassa	
	Pvm	Lainaja	Puhelin
1			
	18.3.03	Juha Taina	44 226
	31.3.03	Inkeri Verkamo	44 219
	15.5.03	Hannu Erkiö	44 253
1			
	10.6.03	Petri Myllymäki	44 173
	12.7.03	Mikael Jokela	44 628

Omat tilaukseni	
Tekijä(t)	Kirjan nimi
Card Stuart, Mackinlay Jack, Shneiderman Ben	Readings in Information Visualization: Using Vision to Think
	Tänään ma 10.11. klo 15 mennessä osoitteeseen Teollisuuskatu 23

Sari A. Laakso (2003)

Lähteitä

- Bias91 Bias R. G.,
Walkthroughs: efficient collaborative testing.
IEEE Software, Vol. 8, No. 5, 1991, s. 94-95.
- Broadbent75 Broadbent D.,
The magic number seven after twenty years.
Teoksessa Kennedy R., Wilkes A. (toim.), Studies in long term
memory. Wiley, New York, 1975, s. 253-287.
- Card99 Card S. K., Mackinlay J., Shneiderman B. (toim.),
Readings in Information Visualization: Using Vision to Think.
Morgan Kaufmann, San Francisco, CA, USA, 1999.
- Carroll94 Carroll J. M.,
Making Use A Design Representation.
Communications of the ACM, Vol. 37, No. 12, December 1994, s. 29-
35.
- Carroll00 Carroll J.M.,
Making Use. Scenario Based Design of HumanComputer Interactions.
The MIT Press, USA, 2000.
- Cato01 Cato J.,
User-Centered Web Design.
Pearson Education, 2001.
- Cooper95 Cooper A.,
About Face: The Essentials of User Interface Design.
IDG Books, USA, 1995.
- Cooper03 Cooper A., Reimann R. M.,
About Face 2.0. The Essentials of Interaction Design.
Wiley Publishing, Inc., Indianapolis, Indiana, 2003.
- Cowan01 Cowan N.,
The magical number 4 in short-term memory: A reconsideration of
mental storage capacity.
Behavioral and Brain Sciences, 24, 2001, s. 87-185.
www.bbsonline.org/documents/a/00/00/04/46/bbs00000446-00/bbs.cowan.html
- Dubberly87 Dubberly H., Mitch D.,
The Knowledge Navigator.
Apple Computer, videonauha. Esiintyy videolla Myers B. A. (toim),
HCI'92 Special Video Program.
- Go04 Go K., Carroll J. M.,
The Blind Men and the Elephant: Views of Scenario-Based System
Design.
interactions, November/December 2004, s. 44-53.

- Goldstein02 Goldstein E. B.,
Sensation and Perception.
Wadsworth Publishing, 6th edition, 2002.
- Hackos98 Hackos J., Redish J. C.,
User and Task Analysis for Interface Design.
John Wiley & Sons, USA, 1998.
- Hornbæk06 Hornbæk K.,
Current practice in measuring usability: Challenges to usability studies
and research.
International Journal of Human-Computer Studies, Vol. 64, No. 2,
2006, s. 79-102.
- ISO9241-11 International Organization of Standardization, Ergonomic
Requirements for Office Work with Visual Display Terminals (VDTs),
Part 11: Guidance on usability (ISO 9241-11:1998), 1998.
- Jacobson92 Jacobson I.,
Object-oriented software engineering: a use case driven approach.
Addison-Wesley, 1992.
- Kolehmainen06 Kolehmainen A.,
Ohjelmistolle asetettavien käytettävyyksvaatimusten määrittely.
Pro gradu –tutkielma, Helsingin yliopisto, tietojenkäsittelytieteen laitos,
sarja C-2006-46, 2006.
- Laakso00 Laakso S. A., Laakso K.-P., Saura A.,
Improved Scroll Bars.
ACM CHI 2000 conference proceedings, Extended Abstracts, s. 97-98.
www.cs.helsinki.fi/u/salaakso/papers/CHI2000-Improved-Scrollbars.PDF
- Laakso04 Laakso S. A., Laakso K.-P.,
Hyvän käyttöliittymän varmistaminen GUIDe-prosessimallilla.
Helsingin yliopisto, Tietojenkäsittelytieteen laitos, julkaisematon
artikkeli, päivitetty 3.10.2004.
www.cs.helsinki.fi/u/salaakso/papers/GUIDe-suomeksi.pdf
www.cs.helsinki.fi/u/salaakso/papers/GUIDe-suomeksi.html
- Lauesen01 Lauesen S., Harning M.B.,
Virtual Windows: Linking User Tasks, Datamodels, and Interface
Design.
IEEE Software, July/August 2001, s. 67-75.
www.itu.dk/people/slauesen/Papers/VirtualWindowsIEEE.pdf
- Lauesen02 Lauesen S.,
Software Requirements. Styles and Techniques.
Pearson Education Ltd, 2002.
- Lauesen05 Lauesen S.,
User Interface Design. A Software Engineering Perspective.
Pearson Education Ltd, 2005.

- Lewis97 Lewis C., Wharton C.,
Cognitive Walkthroughs.
Teoksessa Helander M., Landauer T., Pradhu P. (toim.), Handbook of
Human-Computer Interaction. Elsevier Science B.V., Netherlands,
1997, s. 717-732.
- Miller56 Miller G. E.,
The Magical Number Seven, Plus or Minus Two: Some Limits on Our
Capacity for Processing Information.
The Psychological Review, 1956, vol. 63, s. 81-97
www.well.com/user/smalin/miller.html
- Nielsen93 Nielsen J.,
Usability Engineering.
Academic Press, New York, 1993.
- Nielsen94 Nielsen J.,
Heuristic evaluation.
Teoksessa Nielsen J., Mack R.L. (toim.), Usability inspection methods.
Wiley, New York, USA, 1994, s. 25-62.
- Nielsen00 Nielsen J.,
Why You Only Need to Test With 5 Users.
Alertbox, March 19, 2000.
www.useit.com/alertbox/20000319.html
- Nielsen03 Nielsen J.,
Usability 101: Introduction to Usability.
Alertbox, August 25, 2003.
www.useit.com/alertbox/20030825.html
- Norman90 Norman D.,
The Design of Everyday Things.
Doubleday, New York, 1990.
(Alkuperäinen painos nimellä *The Psychology of Everyday Things*,
Basic Books, New York, 1988; suomenkielinen versio nimellä *Miten
avata mahdollisuus ovia*)
- Peterson59 Peterson L. R., Peterson M. J.,
Short-term retention of individual verbal items.
Journal of Experimental Psychology, Vol. 58, 1959, s. 193-198.
- Preece94 Preece J.,
Human-Computer Interaction.
Addison-Wesley, Wokingham, UK, 1994.
- Rubin94 Rubin J.,
Handbook of Usability Testing: How to Plan, Design, and Conduct
Effective Tests.
John Wiley & Sons, 1994.
- Sears97 Sears A.,
Heuristic Walkthroughs: Finding the Problems Without the Noise.
International Journal of Human-Computer Interaction, Vol. 9, No. 3,
1997, s. 213-234.

- Shneiderman98 Shneiderman B.,
Designing the User Interface.
Addison-Wesley, 3rd edition, 1998.
- Shneiderman00 Shneiderman B., Kang H.,
Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos.
International Conference on Information Visualisation (IV2000),
London, England, 2000.
www.cs.umd.edu/hcil/photolib/paper/DirectAnnotation7.doc
- Tufte83 Tufte E. R.,
The Visual Display of Quantitative Information.
Graphics Press, USA, 1983.
- Tufte90 Tufte E. R.,
Envisioning Information.
Graphics Press, USA, 1990.
- Ware00 Ware C.,
Information Visualization: Perception for Design.
Morgan Kaufmann, San Francisco, CA, 2000.
- Williamson92 Williamson C., Shneiderman B.,
The dynamic HomeFinder: Evaluating dynamic queries in a real-estate
information exploration system.
ACM SIGIR'92 conference proceedings, 1992, s. 338-346.