

**JSS Suunnittelu**


- Useat järjestelmät voidaan jakaa kolmeen kerrokseen:
  - käyttöliittymä
    - ulospäin näkyvä osa, palvelujen ulkoasu
  - sovelluslogiikka
    - palvelujen sisältö
  - tietojen säilytys
    - tietokanta, tiedostot

**JSS Suunnittelu**

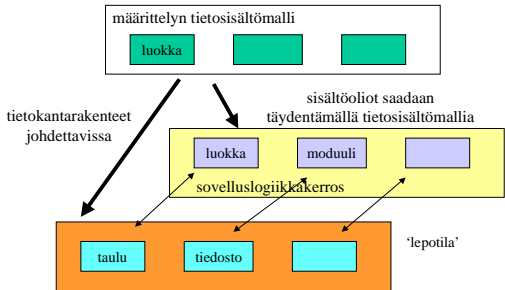
- **Käyttöliittymäkerroksessa käyttöliittymäoliot** (ikkunat, näytöt, www-sivut, valikot, napit, ...) hoitavat yhteyttä käyttäjään esittämällä tietoja ja vastaanottamalla käyttäjän ohjausta.
- **Sovelluslogiikkakerroksessa sisältöoliot** toteuttavat 'reaalimaailman simulointimallin':
  - Sisältöoliot tarjoavat omaan tietosisältönsä perustuvia sovelluskohtaisia palveluja.

**JSS Suunnittelu**

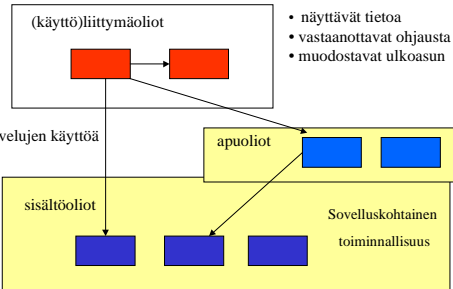
- **Säilytyskerroksessa tietokantaoliot** (tiedostot, relaatiotietokannan taulut, ...) säilyttävät sisältöolioden tiloja.



**JSS Suunnittelu**



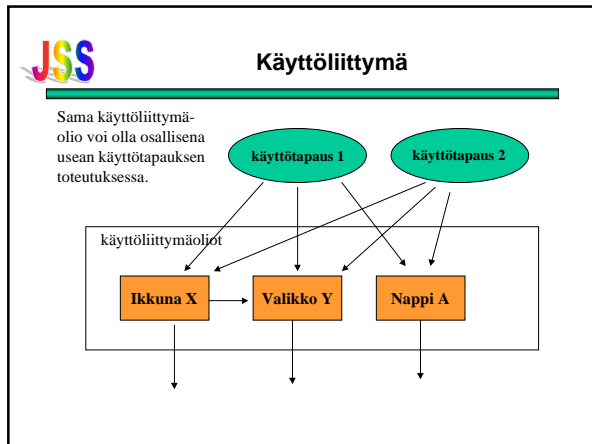
**JSS Suunnittelu**



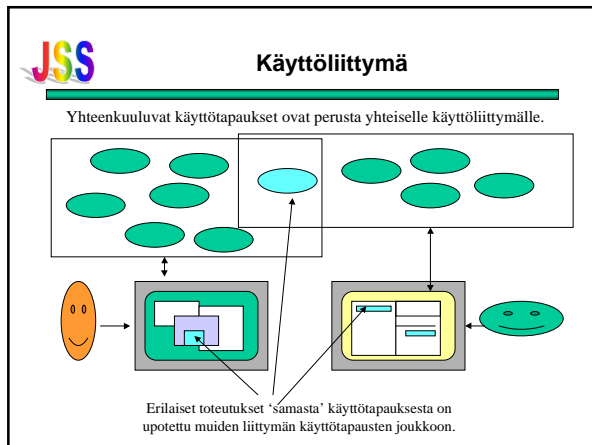
- näyttävät tietoa
- vastaanottavat ohjausta
- muodostavat ulkoasun

**JSS Käyttöliittymä**

- Käyttöliittymän suunnittelussa pyritään **etenemään suoraan käyttäjän tavoitteeseen**.
  - Tavoitteet kuvataan käyttötapausten yhteydessä.
  - Esimerkitapaukset ovat erittäin hyödyllisiä selventämään tavoitteita.
  - Käyttötapausten yhteydessä ollaan kiinnostuneita lähinnä liiketoimintaan liittyvistä tavoitteista.



- JSS Käyttöliittymä**
- Käyttötapausten paketointi yhteisellä käyttöliittymällä toteutettaviksi osajärjestelmiksi on lähtökohta käyttöliittymäsuunnittelulle.
  - **Perusteita paketoinnille:**
    - Käyttäjän tulee saada kaikki tiettyyn asiakokonaisuuteen liittyvät palvelunsa saman käyttöliittymän kautta.
    - Käyttöliittymässä ei pidä olla elementtejä, joita käyttäjä ei tarvitse tai ei voi käyttää.



- JSS Käyttöliittymä**
- Samaan osajärjestelmään otetut käyttötapaukset eivät välttämättä ole aina teknisesti helposti sovitettavissa yhteen.
  - Joudutaan tekemään kompromisseja.
  - Tavoitteille tulisi määritellä tärkeysjärjestys (prioriteetti):
    - auttaa valintatilanteissa

- JSS Käyttöliittymä**
- Toteutusalueen valinta:
    - *laitteisto*
    - *käyttöjärjestelmä*
    - *ikkunointiympäristö*
    - *apuohjelmat, esim. WWW-selain*
  - Ympäristön valinta rajoittaa mahdollisuuksia.
  - Onko sovittava yhteen vai moneen ympäristöön?
  - Mitä useampia alustoja käytetään, sitä vähäisemmät mahdollisuudet.

- JSS Käyttöliittymä**
- Järjestelmän **käytettävyys** (usability): kuinka hyvin käyttäjät pystyvät käyttämään järjestelmää tarkoitettuihin tehtäviin
  - Käytettävyteen vaikuttaa siis
    - mitä järjestelmällä voi tehdä
      - *tiedot, palvelut*
    - millainen käyttäjä on
      - *kokemus, käsitteistö*
    - mitä tehtäviä on tarkoitus suorittaa
      - *erilliset tehtävät, tehtäväkokonaisuudet*

**JSS** **Käytettävyys**

- Käytettävyys on läheisesti yhteydessä käyttöliittymän ominaisuuksiin, mutta muutkin järjestelmän ominaisuudet vaikuttavat siihen.
- Käytettävyyden osatekijöitä:
  - **opittavuus** (learnability)
  - **tehokkuus** (efficiency)
  - **muistettavuus** (memorability)
  - **virheettömyys**
  - **tydydyttävyys** (satisfaction)

**JSS** **Käytettävyys / opittavuus**

- **Onko järjestelmää helppo oppia käyttämään?**
- Järjestelmillä erilaisia käyttäjiä:
  - aloittelijat – rutinoituiden käyttäjät
  - satunnaiset – säännöllisesti käyttävät
- Opittavuus on erityisen tärkeää aloittelijoille ja satunnaiskäyttäjille.
- Satsaukset opittavuuteen eivät saisi kuitenkaan haitata säännöllisesti käyttäviä.

**JSS** **Käytettävyys / opittavuus**

- Opittavuuden saavuttaminen:
  - Perusta käyttöliittymä käyttäjien termistöön, ei atk-termeihin.
  - Noudata standardeja.
  - Matki = tee niin kuin muutkin tekevät.
    - Käyttäjät käyttävät muitakin ohjelmia ja oppivat niistä.
    - Poikkeamat vallitsevasta tavasta häiritsevät opittavuutta.
  - Tee käyttöliittymästä intuitiivinen:
    - liittymä, joka houkuttelee tekemään oikein

**JSS** **Käytettävyys / opittavuus**

- Opittavuuden saavuttaminen:
  - Tee käyttöliittymästä mahdollisimman yksinkertainen.
    - Usein tehtävät asiat pitää voida tehdä suoraviivaisesti, ei monimutkaisia toimintasarjoja.
    - Ota mukaan vain välttämättömiä asioita.
  - Jos vaihtoehtoja on rajattu määrä, tarjoa käyttäjälle mahdollisuus valita.
    - Valitseminen on helpompaa kuin vaihtoehtojen muistaminen.
    - On helpompi tunnistaa oikea vaihtoehto kuin muistaa, miten se olisi kirjoitettava.

**JSS** **Käytettävyys / opittavuus**

- valintatapoja:
  - **poissulkevat valintanapit (radio button)**
    - pieni määrä vaihtoehtoja, aina yksi valitaan
      - vaihtoehto 1  vaihtoehto 2
  - **valintaruudut (checkbox)**
    - pienehkö määrä vaihtoehtoja, voi valita 0 tai useampia
      - vaihtoehto 1  vaihtoehto 2
      - vaihtoehto 3  vaihtoehto 4

**JSS** **Käytettävyys / opittavuus**

- valintatapoja:
  - **valikot (menus)**
    - useita vaihtoehtoja, vievät vähemmän tilaa kuin valintanapit ja ruudut, vieritysmahdollisuus
    - soveltuvat parhaiten tilanteisiin, joissa valitaan yksi vaihtoehto
    - useiden vaihtoehtojen valinta valikosta ei ole välttämättä ilmeistä käyttäjälle eivätkä käyttäjät osaa käyttää valikoita monivalintatilanteissa
    - valikon vaihtoehtoilta on oltava jokin mielekäs järjestys
    - hierarkkiset valikot ja sivuvalikot saattavat joskus kätkeä vaihtoehtoja ja vaikeuttaa niiden löytämistä, erityisesti keinotekoisia hierarkioita pitäisi välttää

**JSS Käytettävyys / opittavuus**

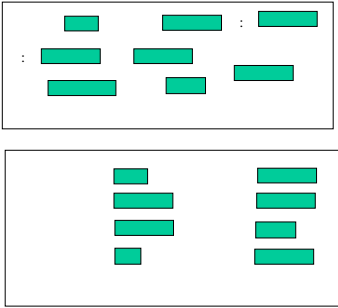
– valintatapoja

- *kuvakkeet (ikonit, icons)*
  - Kuvakkeet ovat pieniä vaihtoehdolle annettuja kuvasympboleja.
  - Kuva saattaa viedä vähemmän tilaa kuin teksti.
  - Kaikille asioille ei ole helppo löytää havainnollisia kuvia.
  - Kuvasympolin merkitys vaatii aina opettelua.
  - Kuvakkeiden ohessa olisi hyvä käyttää vihjetekstejä, jotka ilmestyvät näkyviin, kun osoitin on ollut jonkin aikaa symbolin päällä.

**JSS Käytettävyys / opittavuus**

- Käytä kuvaavia otsaketekstejä ja vaihtoehtojen nimiä.
  - Käytä termejä yhdenmukaisesti, kullekin asialle vain yksi nimi.
- Tee käyttöliittymästä ulkoasultaan selkeä ja sopivan väljä.
  - rivit ja sarakkeet kaaoksen asemasta
- Ryhmittele yhteenkuuluvat asiat omiksi kokonaisuuksikseen.
  - Käytä osien kokoamiseen tyhjää tilaa, värejä, viivoja.

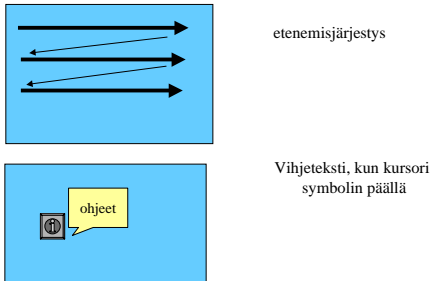
**JSS Käytettävyys / opittavuus**



**JSS Käytettävyys / opittavuus**

- Työskentely etenee luonnostaan normaalin lukemisjärjestyksen mukaisesti:
  - riveittäin vasemmalta ylhäältä oikealle alas
- Tarjoa avustusta:
  - vihjetekstit
- Tarjoa käyttöohjeet on-line avustuksena.
  - Käyttöohjeeseen turvautuminen on äärimmäinen keino, jota käyttäjä tarvitsee, kun käyttöliittymä on epäonnistunut tuomaan esiin, mitä pitäisi seuraavaksi tehdä.
  - Mahdollista asian etsiminen käyttöohjeesta.

**JSS Käytettävyys / opittavuus**



**JSS Käytettävyys / tehokkuus**

- Onko käyttäjän työskentely tehokasta järjestelmän avulla?
- Tehokkuus on usein tärkeä vaatimus: järjestelmällä saatetaan pyrkiä nimenomaan toiminnan tehostamiseen.
- Tehokkuuden tekijöitä:
  - järjestelmän tehokkuus:
    - vasteaika
  - käytön tehokkuus:
    - käyttäjän työmäärä
    - huomion herpaantuminen

**JSS** Käytettävyys / tehokkuus

- Vasteaika (response time):
  - aika joka kuluu käyttäjän toimenpiteestä siihen, että järjestelmä on antanut palautteen
    - 0.1 sekuntia: käyttäjällä tuntuma välittömästä reagoinnista
    - 1 sekunti: käyttäjä huomaa viiveen, mutta se ei vielä ole häiritsevää
    - 10 sekuntia: keskittyminen herpaantuu
      - tarvitaan palaute kertomaan kauanko kestää
      - kokematon käyttäjä voi kuvitella, että järjestelmä on kaatunut

**JSS** Käytettävyys / tehokkuus

- Vasteaikaan vaikuttavia tekijöitä:
  - siirrettävän tiedon määrä
  - tiedon siirtopolun pituus (prosessoinnin määrä)
    - suoraan paikalliselta levyltä vs.
    - tietokantapalvelimelta, www-palvelimen ja edustapalvelimien kautta
    - palvelimien tehokkuus ja kuormitus
  - tietoliikenneyhteyksien nopeus ja ruuhkat
  - esim. WWW-käytössä esiintyy usein häiritsevän pitkiä vasteaikoja pienilläkin tietomäärillä
- Vasteajan vaihtelu lisää häiritsevyyttä.

**JSS** Käytettävyys / tehokkuus

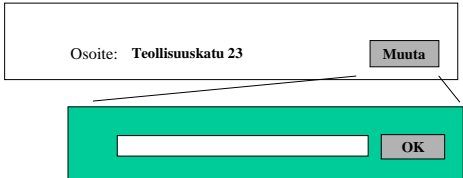
- Miten vasteaikaan voi vaikuttaa:
  - tehokkaat palvelimet ja tietoliikenneyhteydet
  - siirrettävän tiedon määrän minimointi
    - Älä siirrä mitään turhaa, mitä käyttäjä ei juuri siinä tilanteessa tarvitse.
    - Vältä kuvia (etenkin jos siirtotie on hidas).
- Joskus liian nopea vastekin voi olla ongelma!
  - Teksti vilahtaa ja katoaa eikä käyttäjä ehdi lukea sitä.

**JSS** Käytettävyys / tehokkuus

- Käyttäjän ajankäytön tehokkuus on vähintään yhtä keskeistä kuin järjestelmän tehokkuus:
  - käyttäjän miettimisaikaa mitataan minuuteissa, järjestelmän suoritusaikaa sekunnin murto-osissa!
- Käyttäjän työmäärää voidaan mitata
  - näppäinpainallusten määrällä,
  - hiiren liikkeiden määrällä,
  - kirjoitettavan tiedon määrällä,
  - tehtävään kuluvalle ajalle.

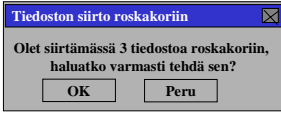
**JSS** Käytettävyys / tehokkuus

- Työmäärään voi vaikuttaa
  - karsimalla turhia tehtäviä, kuten
    - ylimääräisen editointi-ikkunan avaaminen ja sulkeminen tiedon muokkausta varten



**JSS** Käytettävyys / tehokkuus

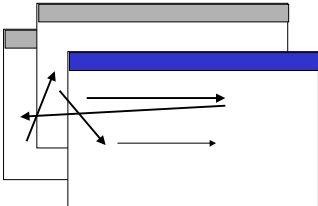
- Työmäärään voi vaikuttaa
  - karsimalla turhia tehtäviä, kuten
    - Tarpeettomat kuitaukset



- Ongelmana on paitsi tarpeeton työvaihe, myös vastauksen muuttuminen automaattitoiminnaksi ja käyttäjän ärsyntyminen.

**JSS** **Käytettävyys / tehokkuus**

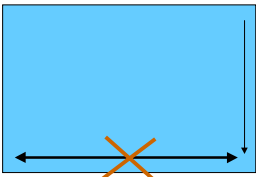
- Työmäärään voi vaikuttaa
  - karsimalla turhia tehtäviä, kuten
    - Käyttäjän pakottaminen hyppimään ikkunoiden välillä



**JSS** **Käytettävyys / tehokkuus**

- Työmäärään voi vaikuttaa
  - pitämällä syötetty tieto muistissa ja käyttämällä sitä:
    - Älä kysy käyttäjältä samaa tietoa kahdesti.
    - Älä kysy mitään, minkä järjestelmän pitäisi tietää.
      - mikä päivä tänään on?
  - minimoimalla työvaiheiden esim.näppäilyjen ja hiiritoimintojen määrää
    - elementtien sijoittelu - yhteenkuuluvat lähellä toisiaan
    - sopiva ikkunakoko - kaikilla ei ehkä ole 20" näyttöä
    - vierityksen välttäminen

**JSS** **Käytettävyys / tehokkuus**



myös pystyvieritystä tulisi välttää, ainakin edestakaista

erityisesti vältettävä vaakavieritystä

Jos käsiteltävänä on esim. iso kuva, ei vieritystä voi välttää; entä automaattivieritys?

**JSS** **Käytettävyys / tehokkuus**

- Työmäärään voi vaikuttaa
  - tarjoamalla oikoteitä (shortcut)
    - pikavalintänäppäilyt valikkovaihtoehdoille
  - reagoimalla virhetilanteisiin välittömästi
    - myöhään tuleva palaute aiheuttaa asiayhteyden vaihtumisen
  - tuomalla palvelut tarjolle selkeästi ja yksinkertaisesti
    - käyttäjän pitää löytää etsimänsä asiat nopeasti

**JSS** **Käytettävyys / tehokkuus**

- Huomion herpaantuminen aiheuttaa **asiayhteyden vaihtumisen** ja vaatii uudelleen palauttamisen, kun työskentelyä jatketaan.
- Herpaantumisen voi aiheuttaa
  - liian pitkä vasteaika
  - pop-up ikkunat ja viestit
    - vaativat käyttäjää keskittymään viestiin
  - näytöllä tapahtuvat muutokset
    - vilkkuvat tekstit, vaihtuvat kuvat, animaatiot, mainokset, piippaukset



**JSS** **Käytettävyys / muistettavuus**

- Onko käyttäjän helppo muistaa, miten järjestelmää käytetään?
- Muistettavuuteen voi vaikuttaa samantapaisesti kuin opittavuuteen, esim.
  - yksinkertaiset ja selkeät toimintaketjut
  - vihjetekstit
  - älä kirjoita vaan valitse
  - yhdenmukaisuus
  - mukautuminen käyttäjän tarpeisiin - käyttäjäprofiilit
    - asetusten muuttaminen kertaalleen pitäisi riittää

**JSS Käytettävyys / virheettömyys**

- Toimiiko järjestelmä virheettömästi oikein annetuilla syötteillä?
  - Tämä riippuu sekä järjestelmän sovelluslogiikan toteutuksesta että käyttöliittymästä.
- Entä 'virheelliset' syötteet?
  - Käytettävyyden kannalta parasta olisi pyrkiä käyttöliittymään, jossa ei voi antaa virheellisiä syötteitä.

**JSS Käytettävyys / virheettömyys**

- valitseminen kirjoittamisen asemesta
  - tarjotaan valittavaksi vain hyväksyttävät vaihtoehdot (epäkelvot harmaiksi)
    - Miksi annettaisiin käyttäjän tehdä valinta ja sitten ilmoitetaan, ettei noin saa tehdä?
  - Rajattua arvojoukkoa käsittelevät kontrollit
    - liukusäätimet,  

    - kasvatettavat syöttökentät  


**JSS Käytettävyys / virheettömyys**

- virheiden estäminen:
  - Ei kysytä asioita, jotka järjestelmän pitäisi tietää.
  - Arvataan oikea vaihtoehto (esim. isot vs. pienet kirjaimet)
  - Hyväksytään vaihtoehtoisia syötteitä - väljennetään syntaksivaatimuksia.
  - Automaattinen täydennys
  - Automaattinen muotoilu
  - Kentiin täyttöohjeet
    - Toimituspäivä (pp.mm.vvvv): \_\_\_\_\_
  - Numeerinen kenttä hyväksyy vain numeerista dataa ja reagoi välittömästi.
  - Onko muita tapoja syöttää tieto virheettömästi?
    - uudet mediat

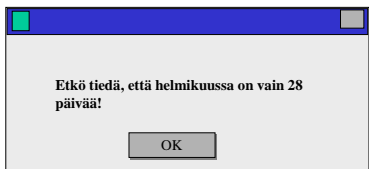
**JSS Käytettävyys / virheettömyys**

- Käyttäjät tekevät kirjoitusvirheitä ja jopa ajatusvirheitä. Kaikkia virheitä ei käytössä-olevalla tekniikalla voi estää. Jos syötettävä tieto on välttämätön toiminnan jatkamisen kannalta, on virheestä ilmoitettava käyttäjälle.
  - pankkitilin numero ja rahasumma tilisiirrossa
  - opiskelijanumero ilmoittautumisessa

**JSS Käytettävyys / virheettömyys**

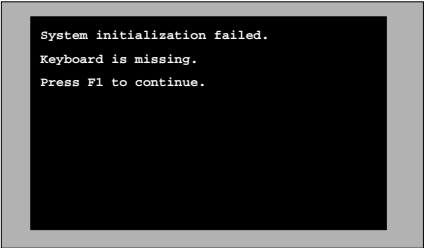
- Miten ilmoitetaan virheestä?
  - ei välttämättä mitenkään
    - Jos oikein syötetystä ilmoitetaan vaikkapa vaihtamalla kentän väri tai jos oikein tehty operaatio tuottaa muun näkyvän tuloksen, positiivisen kiittauksen puuttuminen ilmaisee käyttäjälle, että jotain meni pieleen.
  - opasteella kentän vieressä
  - ellei muuta keinoa löydy niin viesti-ikkunalla
    - ilmoituksen tulisi olla opastava ja asiallinen,
    - ei loukkaava, syyttävä, 'hauska', typerä

**JSS Käytettävyys / virheettömyys**



Tyhmä käyttäjä!

**JSS** Käytettävyys / virheettömyys



Mitenkähän se onnistuu?

**JSS** Käytettävyys/virheettömyys

- Virheestä on syytä ilmoittaa **heti** eikä vasta ikkunaa suljettaessa tai lomaketta lähetettäessä.
- Peruutusmahdollisuus eliminoi vakavat virheet.
  - *rajaton monen askeleen peruutus parempi kuin yhden askeleen*
- Automaattinen tilan tallennus estää tehdyn työn mitätöitymisen.
- Lopetettaessa tehtävä automaattitallennus edellyttää, että vanha versio jää talteen.
- Versiointi päälletallennuksen asemesta lisää muutenkin turvallisuutta.

**JSS** Käytettävyys / virheettömyys

- Liian lähellä toisiaan olevat operaatiot ovat vaarallisia.
  - *esim. unix:ssa*
    - rm – poista tiedosto
    - em – käynnistä emacs editori

**JSS** Käytettävyys / tyytyväisyys


- **Kokeeko käyttäjä järjestelmän käytön miellyttäväksi?**
- Suurimman osan tässä osuudessa käsiteltävistä asioista olisi aivan yhtä hyvin voinut käsitellä tehokkuuden yhteydessä.
- Tyytyväisyyteen vaikuttavat järjestelmän tai käyttöliittymän ominaisuuksien lisäksi myös käyttäjän tottumukset, arvostukset, jne.

**JSS** Käytettävyys / tyytyväisyys

- **Värien käyttö**
  - Värejä voidaan käyttää
    - *asioiden ryhmittelyyn,*
    - *huomion kiinnittämiseen,*
    - *houkuttelevuuden lisäämiseen,*
    - *tyylikkyyden tavoitteluun, yms.*
  - Väreihin liittyy vakiintuneita mielleyhtymiä, joita voi hyödyntää:
    - **vihreä** on sallittu, hyväksyttävä
    - **punainen** on kielletty, varottava

**JSS** Käytettävyys / tyytyväisyys

- Jotkut värit eivät sovi yhteen, esim.



pinkki teksti vihreällä taustalla

vihreä teksti sinisellä taustalla

heikko kontrastiero taustan ja tekstin välillä



**JSS Käytettävyys / tyytyväisyys**

- Liian paljon värejä näytöllä johtaa värikaakseen:
  - näyttö on sekava ja levoton
  - 4-5 väriä maksimi
  - räikeät värit ärsyttävät monia
- Kaikki eivät näe / erota värejä yhtä hyvin.
- Ikkunan / www-sivun taustan pitäisi mieluiten olla yksivärinen tai hyvin heikosti kuvioitu.
- Taustan ja tekstin välillä pitää olla riittävän suuri kontrastiero.

**JSS Käytettävyys / tyytyväisyys**

- Fonttien käyttö
  - johdonmukaisuus: samat asiat samalla fontilla
  - ei liikaa fontteja (koko + leikkaus), levotonta
  - max 3-4

Teksti joka sisältää runsaasti erilaisia fontteja, yleensä pikemminkin ärsyttää kuin ilahduttaa.

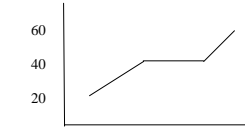
**JSS Käytettävyys / tyytyväisyys**

- Käyttäjän muistia ei pitäisi rasittaa:
  - Käyttäjän työmuistin kapasiteetti on rajallinen (noin 7 objektia).
  - Suuri elementtien määrä johtaa työmuistin ylivuotoon.
  - Operaation selkeä päättymisen vapauttaa työmuistin.
  - Kuvan saattaa hahmottaa yhtenä objektina, lukujoukon luvut kunkin eri objektina.

**JSS Käytettävyys / tyytyväisyys**

- Asioiden näyttäminen havainnollisesti diagrammeina, kuvina, yms. on selkeämpää ja nopeammin hahmotettavaa kuin niiden esittäminen numeroina.

a	20
b	40
c	40
d	60

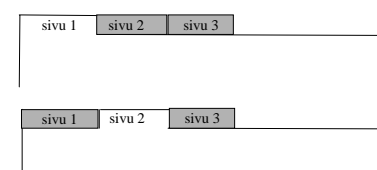


**JSS Käytettävyys / tyytyväisyys**

- Työskentely käyttöliittymän kanssa saattaa pysähtyä erilaisiin keskeytyksiin, esim. puhelinsoittoon.
- Kun työskentelyä jatketaan, on aluksi hahmotettava ja palautettava mieleen tilanne, jossa oltiin.
  - Käyttöliittymän tulisi tarjota tukea tälle.
  - Missä ollaan, mitä ollaan tekemässä?

**JSS Käytettävyys / tyytyväisyys**

- Tekniikkoja käsittelytilanteen hahmottamiseen:
  - lipukkeet (tabs)



**JSS** **Käytettävyys / tyytyväisyys**

- Lipukkeita käytettäessä nähdään
  - kaikki vaihtoehdot
  - missä nyt ollaan
  - kyseessä on siis koko ajan näkyvä valikko, joka näyttää valitun vaihtoehdon
  - valikoiden suunnittelusäännöt

**JSS** **Käytettävyys / tyytyväisyys**

- Hierarkkiset lipukkeet

Luennot	Aukeaa kun Materiaali valitaan
Materiaali	
Kalvot	
Moniste	
Oheismateriaali	
Harjoitukset	

**JSS** **WWW-sivuista**

- Seuraavia asioita pitäisi välttää www-sivujen laadinnassa (Nielsen):
  - vanhentunut tietosisältö
  - virheet
  - hidas latautuminen
  - liikkuva teksti ja animaatioilmukat
  - epästandardit linkit
  - navigointituen puute
    - käyttäjä ei tiedä missä ollaan

**JSS** **WWW-sivuista**

- mutkikkaat URL:t
- kehykset:
  - *suurin ongelma on, ettei alisivuun pysty viittaamaan*
- orvot sivut:
  - *sivulla ei ole linkkejä eikä paluumahdollisuutta*
- vieritettävät linkkilistat:
  - *käyttäjää ei etene loppuun*

**JSS** **WWW-sivuista**

- Lisää vältettäviä asioita (Nielsen):
  - back-nappulan toiminnan estäminen tai hidastus
  - uusien ikkunoiden avaus
  - käyttöliittymäkomponenttien epästandardi käyttö
  - tekijätietojen puute
  - arkistojen puute
  - ohjaus toiseen osoitteeseen
  - otsikot jotka eivät liity sisältöön
  - huonot hakusanat
  - hitaat palvelimet
  - kaikki mainokselta näytävä

**JSS** **WWW-sivuista**

- Kaikilla WWW-sivuilla pitäisi olla (Nielsen):
  - nime ja logo jokaiselle sivulle, logo toimii linkkinä kotisivulle
  - etsintämahdollisuus, jos sivustossa on yli 100 sivua
  - suoraviivaiset ja yksinkertaiset otsakkeet sekä sivunimet
  - silmäilyä tukeva esitystapa
    - *oleelliset asiat löytyvät heti yhdellä silmäyksellä*
  - hypertekstin hyväksikäyttö
    - *pois lineaarisesta esityksestä*



### WWW-sivuista

---

- tuotteen kuvat
  - *jos kyseessä on tuote-esittely*
- kuvien rajaus vain oleelliseen ja tärkeään
- linkkivihjeet, jotka kertovat mihin linkki johtaa
- lukemisen mahdollistaminen myös käyttörajoitteisille
- yhdenmukainen esitystapa muiden sivujen kanssa