

# KÄYTTÄJÄLÄHTÖINEN OHJELMISTOKEHITYS JA -TESTAUS: IHMINEN TEORIASSA JA KÄYTÄNNÖSSÄ

Kurssin TJTA104 luento

**Hannakaisa Isomäki**

**8.2.2016**

**Jyväskylän yliopisto**

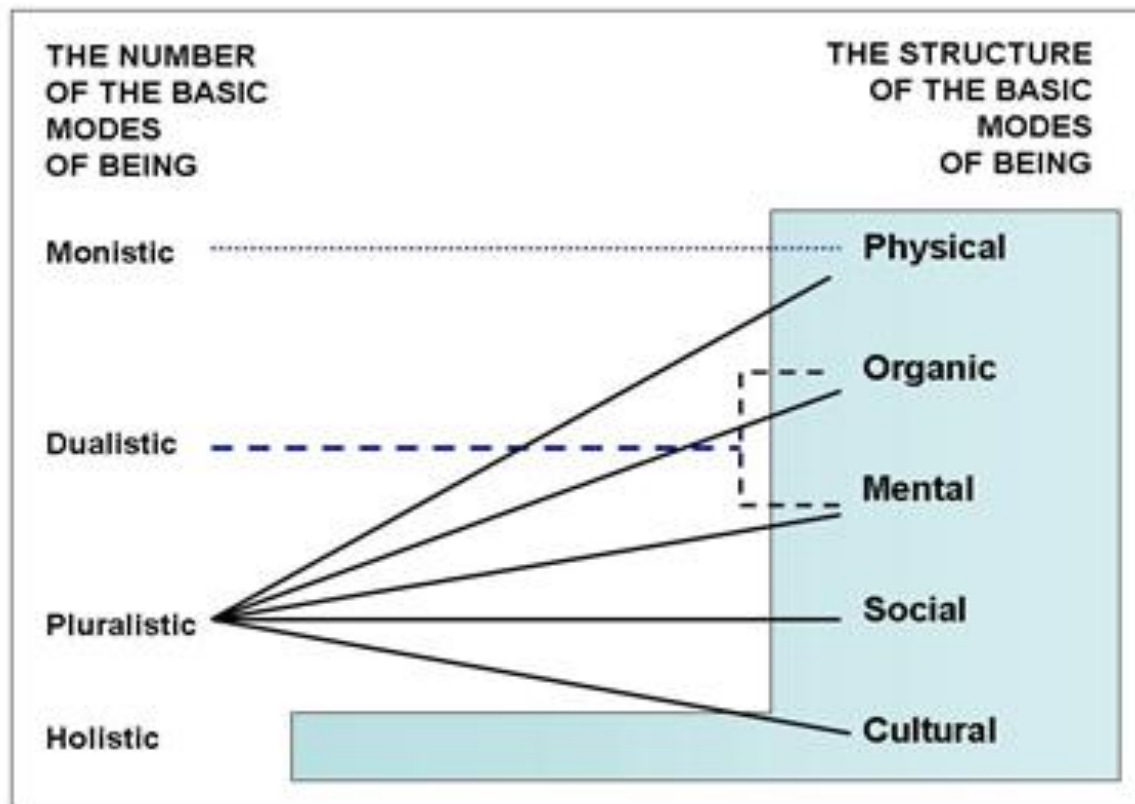
# Käyttäjälähtöinen (KL) ohjelmistokehitys

- ▣ User-centred design (UCD), Information Systems –alan käsite
- ▣ Lähitieteenala: Human-computer interaction (HCI)
- ▣ Ensimmäiset UCD metodit 1980-luvulla:
  - ▣ Soft Systems Thinking eli SSM (Peter Checkland 1981)
  - ▣ ETHICS (Enid Mumford 1983)
  - ▣ Participatory design (Pelle Ehn 1988), skandinaavinen koulukunta
- ▣ Kattoivat koko systeemin kehityskaaren, kehittyessään erikoistumista toisaalta ohjelmistotekniikkaan & asiakasrajapintaan (esim. ketterä kehittäminen) ja toisaalta käyttöliittymien kehittämiseen (räätälöidyt järjestelmät vs. massatuotteet)
- ▣ Käyttöliittymien suunnittelussa huomio käytettävyyden suunnittelussa ja testauksessa: nykyisin tärkeätä yhdistää visuaalisuus, toiminnallisuus ja käytettävyys (vrt. esim. älypuhelimien käyttöliittymät vs. HTML-pohjaiset sivut)

# KL kehittämisen tavoitteita

- Käyttäjien määrittely (Identification of users; esim. ISO3407, ISO9241-11 standardit)
- Käyttäjien tunnistaminen: kenelle ohjelmisto tehdään?
  - Demografiset tiedot
  - Kuluttajakäyttäytyminen
  - Ohjelmiston käyttötavat (mitä käyttäjät haluavat sillä tehdä ja miten)
  - Missä käyttäjät ohjelmistoa käyttävät (fyysinen ympäristö)
  - Keiden kanssa käyttäjät ovat vuorovaikutuksessa ohjelmiston kanssa
  - Miten käyttäjät tulee ottaa mukaan ohjelmiston suunnitteluun
  - **Minkälaisia käyttäjät ovat ihmisinä ? (vaikuttaa kaikkeen edelliseen)**
- Lähde: livari, J., Isomäki, H. & Pekkola, S. 2010. The user – the great unknown of systems development: reasons, forms, challenges, experiences and intellectual contributions of user involvement. Information Systems Journal 20, 109-117. <http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2575.2009.00336.x/full>

# Käyttäjälähtöisen ohjelmistokehityksen ihmiskäsitys eli käyttäjäkokemuksen teoreettiset lähtökohdat (Isomäki 2002; 2009)



Lähde: Isomäki 2002. <http://tampub.uta.fi/handle/10024/67210> (pysyvä: <http://urn.fi/urn:isbn:951-44-5388-3>)

# Tehtävä

- **Muodostakaa 2-3 hengen ryhmiä ja pohtikaa yhdessä näitä kysymyksiä:**
- **Minkälaisina ihmisinä koette itsenne teknologian käyttäjinä?**
- **Mitä seikkoja itsessänne haluaisitte otettavan huomioon kun teknologiaa suunnitellaan käyttäjille?**

# KL-kehittämisen peruselementit: IT-ammattilaisen ajattelu ja KL-menetelmä

- Käyttäjälähtöisyydessä keskeistä on käyttäjän näkökulmasta ohjelmiston ja käyttöliittymän suunnittelu, toteutus ja testaus: on ymmärrettävä ja analysoitava käyttäjän ominaisuuksien kautta ohjelmistoa.
- Tässä apuna sekä IT-ammattilaisten käyttäjälähtöinen ajattelu että KL-menetelmän käyttö osana ohjelmistokehitystä  
**IT-AMMATTILAISEN AJATTELU + KL-MENETELMÄ = HYVÄ DESIGN**
- Käyttäjälähtöisessä ajattelussa tärkeätä erottaa ohjelmistokehityksessä tarvittavat eri näkökulmat sekä miten niitä voi kehittämisessä hyödyntää

# Separatistinen ajattelu: käyttäjä erossa kehittämisestä

## Esimerkki 1

Haastattelija: *"Kenelle olet tekemässä ohjelmistoa?"*

IT-ammattilainen A: *"Teen sitä speksejä vastaan, tässä kehittämisprosessissa mennään aika strukturoidusti ja mun homma on ohjelmoida speksien mukaan"*

## Esimerkki 2

IT-ammattilainen B: *"...on tullut vastaan käyttäjiä, jotka pelkää niin paljon sitä käyttöliittymää etteivät uskalla tehdä sillä juuri mitään, ehkä vaan niitä totuttuja ja turvallisia asioita"*

## Esimerkki 3

Haastattelija: *"Kenelle olet tekemässä ohjelmistoa?"*

IT-ammattilainen B: *"No en ajattele muuta kuin että teen sitä itselleni, sellaiseksi miten itse sitä haluaisin käyttää. En tiedä miten voisin tehdä sitä muiden ehdoilla"*

## Separatistinen ajattelu: soveltuvuus KL-kehittämiseen

Fokus	Hyödyt	Rajoitukset
Ohjelmistotekniikan elementit (vaatimusmäärittelyt, spesifikaatiot)	Huomiota määrityksiin	Riippuvainen alkuperäisten spesifikaatioiden käyttäjälähtöisyydestä
Käyttäjän emotionaaliset esteet	Käyttäjän kokema pelko osoittaa ohjelmassa kohdan, johon pitää kiinnittää huomiota	Oltava mahdollisuus kohdata loppukäyttäjä ohjelmiston testauksessa/ protoilussa
IT-ammattilaisen oma tapa käyttää ohjelmaa	Käyttötapa on aina läsnä ohjelmistokehityksessä	Sulkee pois kaikki muut käyttäjät paitsi IT-ammattilaiset ("I-methodology", "Inscription")

> Separatistinen ajattelu osoittaa ne seikat, jotka sulkevat loppukäyttäjät pois ohjelmistokehityksestä



# Funktionalistinen ajattelu: välineellinen käyttäjänäkemyks

## **Esimerkki 1**

Haastattelija: *"Miten määrittelisit käyttäjän tarpeet?"*

IT-ammattilainen D: *"Asiakkaan tarpeet ovat kustannustehokas informaatioteknologia, joka palvelee loppukäyttäjiä"*

## **Esimerkki 2**

Haastattelija: *"Miten käyttäjät haluaisivat ohjelmistoa käyttää?"*

IT-ammattilainen E: *"Tehokkuus on ehdottomasti tärkein asia. Käyttäjät haluaa saada työnsä tehdyksi mahdollisimman tehokkaasti."*

## **Esimerkki 3**

Haastattelija: *"Miten määrittelisit käyttäjän tarpeet?"*

IT-ammattilainen F: *"No ne alkaa siitä mitä ihmiset tarvitsee työtehtäviensä tekemisessä ja päättyy siihen mitä organisaation prosesseista voi abstrahoida ja käyttää toiminnan kehittämiseen ja kontrollointiin"*

## Funktionalistinen ajattelu: soveltuvuus KL-kehittämiseen

Fokus	Hyödyt	Rajoitukset
Kustannustehokas IT	Huomiota kustannuksiin	Taloudellisuus on KL-näkökulmasta välineellinen tavoite
Tehokkuus työtehtävien suorittamisessa käyttökriteerinä	Huomiota ohjelmiston tehokkuuteen	Ohjelmiston tehokkuus välineellinen tavoite
Organisaation prosessien abstrahointi tavoitteena kontrolli ja toiminnan kehittäminen	Laaja prosessikeskeinen näkökulma kohdejärjestelmään (prosessi = liiketoimintaprosessi, työ/toimintaprosessit, palveluprosessi, tietovirrat jne...)	Käyttäjälähtöisyyden toteutuminen edellyttää kykyä analysoida ja liittää käyttäjäprofiilin sisältämät asiat mukaan prosesseihin

> Funktionalistinen ajattelu tuo esiin ne toiminnallisuudet, joiden suunnittelussa ja toteutuksessa käyttäjien ominaisuudet tulisi huomioida

# Holistinen ajattelu: ihmisläheinen käyttäjänäkemyks

## Esimerkki 1

Haastattelija: *"Miten määrittäisit käyttäjän tarpeet?"*

IT-ammattilainen H: *"Kaikkein tärkeintä ohjelmiston suunnitteluvaiheessa on, että oikeiden käyttäjien todelliset tarpeet on lähtökohtana"*

## Esimerkki 2

IT-ammattilainen I: *"Tyytyväinen käyttäjä on asiakassuhteen jatkuvuuden ta. Jos käyttäjät eivät ole tyytyväisiä toimitettuun ohjelmistoon, siihen tulisi heti reagoida. Ei saisi hoitaa tilannetta niin että jää mielikuva että häivyttiin heti implementoinnin jälkeen niin että takavalot vaan näkyi."*

## Esimerkki 3

Haastattelija: *"Oletko huomannut samanlaisia piirteitä ihmisissä joille teet ohjelmistoja?"*

IT-ammattilainen J: *"No he ainakin tekevät työtään aivan eri tavoilla ja heillä on ihan erilaiset näkemykset asioista vaikka voivat olla samastakin firmasta"*

## Holistinen ajattelu: soveltuvuus KL-kehittämiseen

Fokus	Hyödyt	Rajoitukset
Oikeiden käyttäjien todelliset tarpeet suunnittelun lähtökohta	Tavoitetaan käyttäjien tarpeet	Tarpeiden selvittämisen rajoitteet (aika, raha, menetelmäosaaminen)
Asiakastyytyväisyys	Huomiota asiakkuuden hoitoon ja asiakasrajapintaan	Asiakastyön rajoitteet (aika, raha, projektiosaaminen)
Ihmisten erilaisuus	Huomioidaan käyttäjien yksilöllisyys	Käyttäjälähtöisyyden toteutuminen edellyttää kykyä analysoida käyttäjäprofiilin sisältämät asiat mukaan käyttöliittymiin

> Holistinen ajattelu keskittyy todellisten loppukäyttäjien oikeisiin tarpeisiin kokonaisvaltaisesti

# KL-menetelmä: Personas

- Microsoftin kehittämä menetelmä (Jonathan Grudin)
- Käytetty mm. MSN Explorerin, MS Windowsin, MS Wordin kehittämisessä
- Tavoitteena luoda käyttöliittymiä, jotka miellyttävät satoja tuhansia/miljoonia käyttäjiä (massatuotteet)
- Kehitetty parantamaan käyttäjälähtöistä skenaariopohjaista suunnittelua: vuorovaikutus ja käyttäjäkokemus suunnitellaan luomalla todelliseen käyttäjätietoon perustuva fiktiivinen käyttäjäprofiili
- Personas-menetelmässä yhdistetään useita eri tietolähteitä (esim. asiakastieto, demografiset tiedot) ja tutkimusmenetelmiä (esim. etnografia, kuluttajatutkimus, haastattelut, markkina-analyysit, fokusryhmät, etc.)

Personas pyrkii parantamaan aikaisempia (Microsoftin käyttämiä) menetelmiä:

- Miten parhaiten voi luoda käyttäjäprofiileja (ilman oikeiden käyttäjien osallistamista ohjelmistoprojektiin)?
- Mikä tieto ja aineisto on soveltuvinta tarkoitukseen?
- Miten kannattaa yhdistää eri aineistoja?
- Miten voi testata käyttäjäprofiilin paikkansa pitävyyden?
- Miten useampi kehittämisprojekti voi jakaa samat käyttäjäprofiilit?
- Paraniko tuote Personas-menetelmää käyttämällä?

# Personas-menetelmän päävaiheet 1 (3)

- Kerätään paljon jo olemassa olevaa faktatietoa potentiaalisista käyttäjistä
  - ▣ Markkinatutkimukset (massatuote: iso otanta), markkinasegmentit ja kilpailutilannetieto
  - ▣ Käyttäjätutkimukset (etnografiat, fokusryhmät, haastattelut, asiakastiedot (vrt. asiakaskortit, kuittirivitieto))
  - ▣ sekä yrityksen sisäinen että ulkoinen tieto
- Luodaan alustavat käyttäjäprofiilit (3-6 riippuen tuotteen potentiaalisten käyttäjien määrästä)
- Aloitetaan käyttäjätutkimukset ja kerätään lisää tietoa täsmentämään alustavia käyttäjäprofiileja
  - ▣ Esim. fokusryhmät (Windows: 5000 hengen paneeli), haastattelut, havainnointi ym. kvalitatiivinen ja kvantitatiivinen tutkimus

# Personas-menetelmän päävaiheet 2(3)

- Analysoidaan kerätty tieto ja kirjoitetaan käyttäjäprofiili
  - Tutkimustulokset tulkitaan ja yhdistetään käyttäjäprofiiliksi
  - Voidaan luoda erilaisia profiileja: peruskäyttäjä, kansainvälinen käyttäjä, vammainen käyttäjä, anti-käyttäjä (ei kohteena)
  - Narratiivinen tyyli: tarkoitus on tehdä käyttäjäprofiilista mahdollisimman elävä; mahdollisuus samaistua käyttäjään
  - Tavoitteena perustaa jokainen toteamus/seikka käyttäjäprofiilissa käyttäjätutkimuksissa tuotettuun tietoon: faktaan perustuva fiktiivinen käyttäjäprofiili
  - Käytetään seinätekniikkaa, postereita yms. Visuaalista ja sanallista dataa yhdistäviä dokumentteja (Personas-dokumentit toimivat kommunikoinnin välineenä ohjelmistokehitykseen osallistuvien eri ammattilaisten välillä)
  - Luodaan keskeinen suunnitteludokumentti: Foundation document (perusdokumentti)



# Personas-menetelmän päävaiheet 3(3)

- Perusdokumentin sisältö: käyttäjän keskeiset ominaisuudet
  - perustavan laatuinen tieto käyttäjästä: visiodokumentit, käytön tavoitteet, pelot, epätoivotut tilanteet ja muut tyypilliset aktiviteetit, jotka antavat perusteet tarkennetuille dokumenteille
  - tutkimusraportit yms. referenssitieto
  - kuvat, videot (kehitysryhmä kuvaa käyttäjiä)
- linkitykset tarkennettuihin dokumentteihin (feature scenarios)
  - toiminnallisuudet (feature specs), ”walk-through”-skenaariot, visiot, storyboardit (kuvakäsikirjoitukset), kommentit yms. suunnittelun tarkemmat dokumentit
  - jokaisesta dokumentista selkeä linkitys perusdokumentin siihen kohtaan, jota ko. dokumentti tarkentaa

# Perusdokumentti: esimerkki 1 / 3

- Yleiskuvaus käyttäjästä: Patrick Blakeman (pk-yrittäjä)
  - ▣ Get to know Patrick, his business and family
- A Day in the Life (tyypillinen päivä)
  - ▣ Follow Patrick through a typical day
- Work Activities (työtehtävät)
  - ▣ Look at Patrick's job description and role at work
- Household and leisure activities (koti ja vapaa-aika)
  - ▣ Get information about what Patrick does when he is not at work
- Goals, fears and aspirations (tavoitteet, pelot, toiveet)
  - ▣ Ymmärrä Patrickin kiinnostuksen kohteet elämässään, työssään ja liiketoiminnassaan
- Computer skills, knowledge and abilities (IT-tiedot ja taidot)
  - ▣ Opi Patrickin IT-tiedot ja taidot

# Perusdokumentti: esimerkki 2/3

- Market size and influence (markkinakoko ja vaikutus)
  - Ymmärrä mikä on Patrickin kaltaisten ihmisten merkitys liiketoiminnallemme
- Demographic attributes (demografiset tiedot)
  - Lue avain demografiset tiedot Patrickista ja perheestään
- Technology attributes (IT ominaisuudet)
  - Hanki tuntuma siihen miten Patrick käyttää teknologiaa
- Technology attitudes (IT asenteet)
  - Tutki Patrickin näkemykset teknologiasta, menneet ja tulevaisuus
- Communicating (kommunikaatio)
  - Opi miten Patrick pitää yhteyttä ihmisiin
- International considerations (kansainvälinen aspekti)
  - Ota selvää millainen Patrick on ulkomailla

# Perusdokumentti: esimerkki 3/3

- Quotes (lainaukset)
  - ▣ Kuuntele mitä Patrickilla on sanottavaa
- References (viitteet)
  - ▣ Katso tämän dokumentin lähteet ja tarkennukset
  
- Kun perusdokumentti on valmis, sen mukainen käyttäjäprofiili testataan oikeilla käyttäjillä ja kommunikoidaan muulle kehitysryhmälle