



## Johdatus sovellussuunnitteluun

---

- Yleiskuva olioperustaisesta tietojärjestelmän ja erityisesti ohjelmiston kehittämisprosessista sekä siinä käytettävistä kuvaustekniikoista
- Pääperiaatteet käyttöliittymien suunnittelusta



## Johdatus sovellussuunnitteluun

---

- Sisältö
  - *Johdanto*
  - *Ohjelmiston kehitysprosessi*
  - *Käyttötapaukset -UML*
  - *Olioiden ja luokkien kuvaaminen - UML*
  - *Olioyhteistyön kuvaaminen - UML*
  - *Oliomallinnus*
  - *Käyttöliittymän suunnittelun perussääntöjä*



## Johdatus sovellussuunnitteluun

---

- Luennot
  - 10.6., 12.6., 16.6, 18.6., 23.6. ja 26.6 klo 16-20
  - A217, teollisuuskatu
  - Lehtori Harri Laine (laine@cs.helsinki.fi)
- Www-kotisivu
  - <http://www.cs.helsinki.fi/u/laine/jossu/v03/>



## Kurssin suorittaminen

---

- Osasuorituksina koe ja harjoitukset.
  - Kokeesta saatava maksimipistemäärä on 50 pistettä.
  - Harjoituksista saatava maksimipistemäärä on 10 pistettä. Harjoitukset **eivät** ole pakolliset.
  - Arvostelu tulee perustumaan laitoksella yleisesti käytettyyn asteikkoon, jossa hyväksymisraja on 30 ja arvosananävi on 3 pistettä. Pientä säätelyä asteikon suhteen voidaan tehdä.



## Oppimateriaali

---

- Pääasiallisena materiaalina on kalvo- ja monistemateriaali
  - kalvot ja moniste löytyvät kurssin www-kotisivulta (pdf-muodossa)
  - Monistetta löytyy monistemyynnistä

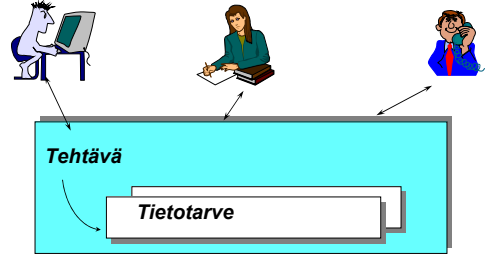


## Oppimateriaali

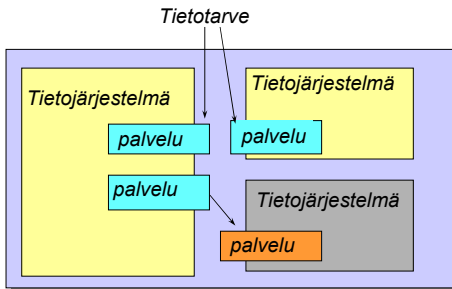
---

- Lähdemateriaalia:
  - *Harri Laine: Johdatus sovellussuunnitteluun, Helsingin yliopisto, tietojenkäsittelytieteen laitos, 2002*
  - *Maciaszek L.A.: Requirements Analysis and System Design, Developing Information Systems with UML, Addison Wesley, 2001*
  - *Booch G., Rumbaugh J. and Jacobson I.: The Unified Modeling Language User Guide, Addison-Wesley, 1998*
  - *Preece J, et al: Human-Computer Interaction, Addison-Wesley, 1994*

- Tällä kurssilla **ei ohjelmoida**, mutta kuitenkin edellytetään
  - Ohjelmoinnin perusteet
  - Olio-ohjelmoinnin peruskäsitteet (luokka, olio, palvelu)
  - Java-koodin välttävä lukutaito (muutamassa harjoitustehtävässä)

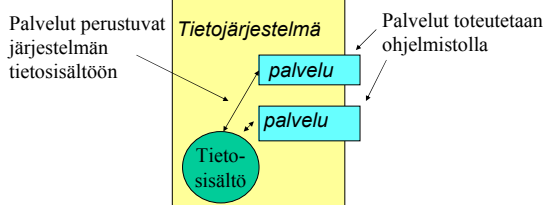


Käyttäjillä esiintyy toiminnassaan (*työ* tehtäviinsään) tietotarpeita = tarpeita käsitellä tietoja

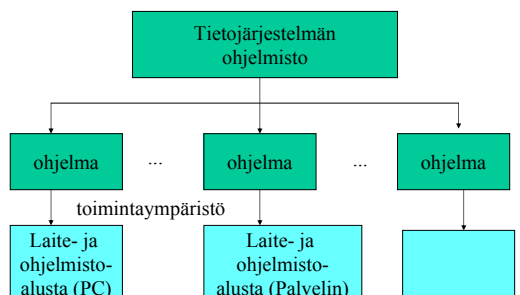


*Tietojärjestelmät tarjoavat palveluita tietotarpeiden tyydyttämiseksi*

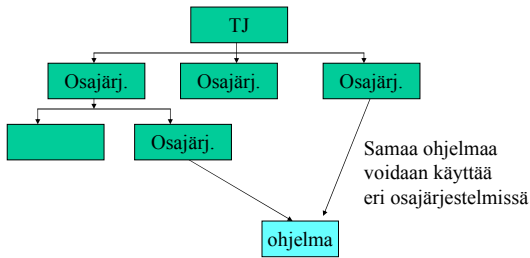
- Henkilö voi tarvita toiminnassaan useiden tietojärjestelmien palveluita
- Tietojärjestelmä on jollain perusteella muodostettu hallinnollinen kokonaisuus



Ohjelmisto koostuu yhdestä tai useammasta jollakin laitteisto- ja ohjelmistoalustalla toimivasta ohjelmasta



Tietojärjestelmä voi jakautua hierarkkisesti osajärjestelmiin



- Sama palveltava toimintakokonaisuus
  - ilmoittautumiset, kurssihallinto
- Yhteiset käyttäjät
  - kirjaston asiakasjärjestelmä
- Yhteinen tietosisältö
  - tilaustietojärjestelmä
- Maantieteellinen sijainti
  - aluekohtaiset asiakasjärjestelmät

- Yhteinen tehtävätyyppi
  - raportointiosajärjestelmä
- Samanlainen ajoitus
  - päivittäistoiminta / vuodenvaihteen toimet
- Yhteinen toteutustekniikka tai -väline
  - puhelinikäyttöinen tilietojärjestelmä
- Osajärjestelmäjako voi olla monikriteerinen ja monitasoinen: eri tasoilla jakoperusteet voivat vaihdella.

- Tietojenkäsittelytieteen laitoksen ilmoittautumisjärjestelmä (ILMO)
- Opintosuoritusrekisteri
- Lentoyhtiöiden paikanvarausjärjestelmä
- Yliopiston kirjaston lainausjärjestelmä
- Puhelinkeskuksen ohjelmisto
- Tekstinkäsittelyjärjestelmä
- Ohjelmointiympäristö
- Tietojärjestelmän kehittämisympäristö (CASE)
- Tietokonepeli

- Sovellusohjelmisto (application software)=
  - Tietojärjestelmän käyttöön tarvittava sovelluskohtaisia palveluita tarjoava yhden tai useamman ohjelman muodostama kokonaisuus. Ohjelmisto voi koostua useilla eri laitteilla toimivista ohjelmista.
- Varusohjelmisto (system software) =
  - sovellusohjelmistolle alustan tarjoava ohjelmisto (käyttöjärjestelmä, ikkunointiympäristö, tietoliikennepalvelut, tietokantapalvelut, ...)
  - Ohjelmiston tuottamishankkeen kannalta oletetaan varusohjelmistopalvelujen olevan valmiina tarjolla

- Mihin tarkoitukseen? (**why**)
  - mikä on järjestelmän tehtävä?
- Millainen? (**what**)
  - mitä ominaisuuksia järjestelmällä on?
  - tietosisältö / palvelut / käyttäjät
- Miten? (**how**)
  - tekninen ratkaisu?



## Järjestelmän tarkoitus

- Millaista toimintaa järjestelmällä pitäisi tukea?
- Mitä tietojenkäsittelytehtäviä järjestelmä avustaa / mahdollistaa?
- Ketkä suorittavat avustettavia tietojenkäsittelytehtäviä?
- Mitä hyötyä / kustannuksia järjestelmästä on?
- Millaisia ongelmia järjestelmällä halutaan poistaa?



## Millainen järjestelmä

- Keitä ovat järjestelmän käyttäjät?
- Mitä palveluja järjestelmä tarjoaa / kenelle?
- Millainen on järjestelmän tietosisältö?
- Millaisessa ympäristössä järjestelmä toimii?
- Millaista on järjestelmän käyttö?



## Miten tehdään

- Millainen on järjestelmän 'tietokannan' rakenne?
- Miten järjestelmää käytetään?
- Millainen käyttöliittymä järjestelmällä on?
- Millainen on ohjelmiston rakenne?
- Miten ohjelmiston osat ovat yhteydessä toisiinsa?



## Järjestelmän elinkaari

- kartoitus
- määrittely
- suunnittelu
- toteutus (ohjelmointi ja testaus)
- käyttöönotto
- ylläpito



## Työmääriä

■ Kartoitus	3 %	9%
■ Määrittely	5 %	15%
■ Suunnittelu	7 %	20%
■ Koodaus	6 %	18%
■ Moduulitestaus	7 %	20%
■ Integrointi	6 %	18%
■ Ylläpito	67 %	

Schach S.R.: Software engineering, 2. ed., Aksen Associates, 1993.



## Kartoitus

- Tarkoitus
- Ongelmat
- Tavoitteet
- Yhteys toimintaan
- Kannattavuus



## Kartoitus (feasibility study)

- Lähtökohtana tilanneselvitys:
  - mikä on tilanne nyt
    - millaisia ongelmia nykytilanteessa
  - mitä odotetaan tilanteen olevan tulevaisuudessa
- Tavoitteiden asetus
  - Tilanneselvitysten perusteella asetetaan tavoitteet tulevalle järjestelmälle
  - tavoitteet tulevilta käyttäjiltä ja järjestelmään liittyviltä **sidosryhmiltä** (= taho, joka on jollain tavoin tekemisissä järjestelmän kanssa)



## Kartoitus / tavoitteiden kohteita

- Tietotarpeiden tyydytys
- Nykyjärjestelmän työnkulun muuttaminen tai muun nykyjärjestelmän ongelman poistaminen
- Uusien palvelujen kehittäminen
- Valvonnan kehittäminen (suojaus)
- Järjestelmän suorituskyky (käsittelymäärät, vasteajat)
- Kustannukset (seuranta, vähentäminen)
- Tehokkuuden parantaminen (käyttöaste)



## Kartoitus / tavoitteiden asetus

- Tavoitteet mahdollisimman täsmällisesti
  - selkeinä numeroina Esim. 'ilmoittautumiseen käytettävä aika on pudotettava 50 % nykyisestä' (Ei: 'ilmoittautumista on nopeutettava')
- Jos tavoitteet liittyvät uusiin palveluihin ja uusiin tietotarpeisiin, on nämä määriteltävä riittävän täsmällisesti.
- Tavoitteet tärkeysjärjestykseen
  - Joskus tavoitteet voivat olla jopa ristiriidassa keskenään



## Määrittely (Requirements analysis)

- Tavoitteiden pohjalta määritellään järjestelmän tavoitetila
- Karkea suunnitelma järjestelmästä =
- Miltä järjestelmä näyttää ulospäin
  - Ketkä järjestelmää käyttävät
  - Mitä palveluja järjestelmä tarjoaa käyttäjilleen
  - Mitä tietoja palvelut tarvitsevat
  - Puiteet järjestelmän toteutukselle (ympäristö, käyttöliittymä,...)



## Määrittely (Requirements analysis)

- Järjestelmä tehdään käyttäjille
- Järjestelmän pitää tarjota ne palvelut, joita käyttäjät haluavat
  - Nämä on siis saatava selville
  - Tietävätkö käyttäjät mitä he haluavat ?
    - Kyllä ja ei:
      - Kysyttävä: Miksi ? Mihin aiot käyttää ?
      - Esiteltävä vaihtoehtoja - Prototyyppi



## Määrittely / vaihtoehtoja

- Kehitetään ratkaisuvaihtoehtoja
  - työnjako (ihminen/kone - palvelu/itsepalvelu)
  - työn muuttuminen <--> tarvittavat palvelut
  - mitä vaikuttaa kokonaisuuteen, jos asiat tehdään tietyllä tavoin
  - käsittelyperiaatteet (suorakäyttö/eräkäyttö)
  - käyttöliittymävaihtoehdot
  - teknisiä vaihtoehtoja, jotka täsmennetään suunnittelussa:
    - tiedostoratkaisu (tiedostot/tietokanta)
    - hajautus (hajautettu/keskitetty)
    - laitteisto ja varusohjelmisto
    - syötteiden ja tulosteiden muoto



## Määrittely / vaihtoehtoja

- Vain jotain tai joitain vaihtoehtoja lähdetään kehittämään eteenpäin, joten on valittava nämä vaihtoehdot
  - vaihtoehtoja on arvioitava
- **Arviointi ?**
  - Tarkastellaan vaihtoehdon vaikutusta
    - käyttäjiin
    - (liike)toimintaan
    - tavoitteiden saavuttamiseen
    - palveluiden saatavuuteen
  - Huom! vaihtoehdot esitettävä konkreettisesti



## Arvioinnissa kiinnitettävä huomiota

- järjestelmän **toimivuuteen**
  - saadaanko kaikki työt hoidettua määritellyllä ratkaisulla ?
  - onko toiminta järkevää ?
  - onko ratkaisu tehokas ?
- järjestelmän **toteutettavuuteen**
  - onko tarvittavat resurssit (henkilöstö, välineet) käytettävissä ?
  - osataanko tehdä (riittävän nopeasti) ?



## Arvioinnissa kiinnitettävä huomiota

- Edistääkö ratkaisu (liike)toimintaa ?
- Onko ratkaisu taloudellisesti kannattava ?
  - Taloudellisuuden selvittäminen: Kustannus-hyöty-analyysi



## Suunnittelu

- Suunnittelussa tarkennetaan määrittelyssä laadittua karkeaa suunnitelmaa. Tarkennus voi tapahtua monitasoisesti asteittain.
- Suunnittelun osa-alueita
  - käyttöliittymän suunnittelu
  - tietokannan suunnittelu
  - työkulkujen suunnittelu
  - laitteiston valinta
  - tietoliikenteen suunnittelu
  - ohjelmistosuunnittelu



## Tietojärjestelmän kuvaaminen

- Tietojärjestelmän kehittämiseen osallistuu yleensä useita henkilöitä
  - on tärkeää, että kehittäjät ymmärtävät toisiaan ja pystyvät tehokkaasti välittämään toisilleen tietoa kehitettävästä järjestelmästä
  - projektiin tulevan uuden henkilön pitäisi pystyä suhteellisen nopeasti muodostamaan kuva järjestelmästä



## Tietojärjestelmän kuvaaminen

- Tietojärjestelmät ovat usein laajoja ja pienehkötkin järjestelmät sisältävät runsaasti yksityiskohtia.
- Yksityiskohtat ovat tärkeitä joissakin tilanteissa, mutta niiden mukanaolo kuvauksessa häiritsee kokonaisuuden hahmotusta
- Tarvitaan siis eri tasoisia kuvauksia, yleiskuvauksista yksityiskohtaisiin kuvauksiin



## Tietojärjestelmän kuvaaminen

- Järjestelmän kuvaamiseen voidaan käyttää luonnollista kieltä (=suomea).
  - Millaisia taloja mahtaisi syntyä, jos niiden suunnitelmat esitettäisiin pelkästään luonnollisella kielellä?
    - *Alenisivatko rakennuskustannukset?*
    - *Saataisiinko paremmin asiakkaan tarpeita tyydyttäviä taloja?*



## Tietojärjestelmän kuvaaminen

- Talonrakennuksen kohdalla on vakiintunut ja jopa viranomaisten säätelemä kuvauskieli, jolla suunnitelmat (piirustukset) laaditaan.
- Tietojärjestelmien kohdalla tilanne on huonompi
  - Tietojärjestelmä on yleensä monimutkaisempi ja sen suunnitelma sisältää enemmän yksityiskohtia kuin talon suunnitelma
  - Tietojärjestelmien kuvauskieliä on esitetty erittäin runsaasti, mutta vakiintunutta rakennuspiirustusten tasoiseen kattavuuden yltävää tekniikkaa ei ole



## Tietojärjestelmän kuvaaminen

- Tietojärjestelmä on abstrakti tuote
- Talonrakennuksen kohdalla on eriytyneitä ammatteja, jotka tarvitsevat omat piirustuksensa. Tietojärjestelmien kohdalla eriytyneisyys on vähäisempää, samat ohjelmoijat käsittelevät kaikkia ohjelman aspekteja käyttöliittymästä tietokantaan
- Ei pystytä teoreettisesti osoittamaan, että jokin kuvaustekniikka olisi parempi kuin toinen
- On erilaisia kilpailevia käsityksiä siitä, millaisista rakenneosista tietojärjestelmä muodostuu, näiden 'hyvyyttä' ei pystytä osoittamaan



## Olioperustainen kuvaaminen

- Tällä kurssilla tarkastellaan järjestelmän olioperustaista kuvaamista
- Olioperustaisuus on nykytrendi ohjelmistotuotannossa. *Tulevaisuudessa ???*
- Kuvausjärjestelmänä esitellään UML-(unified modeling language) kuvaustekniikka.
  - Tekniikkaan sisältyy graafisia kuvauskieliä ja se on saavuttamassa 'teollisuusstandardin' aseman (object management group:n OMG standardoima)
- UML ei ole kaikenkattava kuvausjärjestelmä.