

## Komponenttimallista vielä

- Ohjelmistotuotannon *komponenttimalli* (component-based software engineering (CBSE) perustuu ohjelmistojen uudelleenkäytölle
- *Komponentti* (ohjelmistotuotannon merkityksessä) = itsenäinen ohjelmistoyksikkö, joka tarjoaa palveluitaan hyvin määriteltyjen rajapintojen kautta
  - Tarjottu rajapinta (komponentin tarjoamat palvelut) ja vaadittu rajapinta (palvelut jotka komponentti tarjoaa muille komponenteille)

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 85

## Komponenttimallin oleellisia piirteitä

- Riippumattomat komponentit
  - Rajapinnat
  - Toteutus ja rajapinta selkeästi erotettu
  - Komponentin toteutuksen voi muuttaa ilman järjestelmän muuttamista
- Standardit komponenteille
  - Helpottaa komponenttien integraatiota
- Väliohjelmistot (middleware)
  - Tuki komponenttien yhdessätoimimiselle

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 86

## Vaativuusmäärittely

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 87

## Vaativuusten selvittämisen keskeisyys

- Ehkä kaikkein merkittävin ohjelmistotuotannon ongelma on selvittää asiakkaalta järjestelmälle asetettavat *vaativuudet* (requirements)
- Vaativuudet kuvaavat sen, minkälaisen järjestelmän asiakas haluaa auttamaan tai helpottamaan jotain toimintaa kommunikointiprosessi

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 88

## Vaativuusmäärittelyn tehtävät

- *Vaativuusmäärittely* (requirements engineering) tarkoittaa prosessia, jossa selvitetään järjestelmälle asetettavat vaativuudet
- Vaativuusmäärittelyssä selvitetään
  - mitä järjestelmän pitää tarjota (toiminnot),
  - toivotut ja vaaditut koko järjestelmää koskevat ominaisuudet (laatuvaativuudet),
  - järjestelmän rajoitukset ja reunaehdot

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 89

## Vaativuusten dualistinen luonne

- Yhtäältä vaativuudet tarkoittavat hyvin yleistä kuvausta järjestelmän toiminnasta
- Toisaalta niillä voidaan viitata erittäin yksityiskohtaiseen rakenteeseen esitykseen pienestä järjestelmän yksityiskohdasta
- Molempia ääripäitä tarvitaan, sillä
  - Vaativuusmäärittelyn tuloksia käytetään ohjelmistoprosessin eri vaiheissa
  - Vaativuusmäärittely sisältää informaatiota järjestelmästä eri rooleissa oleville lukijoille

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 90

## Käyttäjä- ja järjestelmävaatimukset

- Vaatimukset jaetaan
  1. *Käyttäjävaa*timuksiin (user requirements)
    - Abstraktimmat, "korkean tason" vaatimukset
  2. *Järjestelmäva*timuksiin (system requirements)
    - Yksityiskohtaisemmat, "matalan tason" vaatimukset

## Vaatimusten kohderyhmät

- Käyttäjä- ja järjestelmävaatimukset pitää erottaa toisistaan, sillä erilaisissa tehtävissä olevat lukijat tarvitsevat erilaista tietoa
- Käyttäjävaa
timusten lukijat ovat usein päälliköitä tai loppukäyttäjiä

  - kiinnostuneita kokonaisuudesta, ei siitä miten kokonaisuus aiotaan yksityiskohdissaan toteuttaa

- Järjestelmävaatimusten lukijat tarvitsevat yksityiskohtia voidakseen
  - suunnitella järjestelmän tai
  - selvittääkseen, miten järjestelmä sopii asiakasyrityksen liiketoimintaan.

## Korkean ja matalan tason vaatimukset

- Korkean tason vaatimuksia käytetään
  - kilpailutettaessa ohjelmistoyrityksiä ohjelmistosopimuksesta
  - kuvattaessa järjestelmää loppukäyttäjille
  - kuvattaessa järjestelmää omalle johdolle
- Matalan tason vaatimuksia käytetään
  - sopimuksena asiakkaan ja ohjelmistoyrityksen välillä
  - suunnittelun lähtökohtana

## Käyttäjävaatimukset

- Kuvataan luonnollisella kielellä ja helposti ymmärrettävillä kaavioilla
  - mitä palveluja järjestelmän odotetaan tarjoavan
  - mitä rajoituksia ja reunaehtoja järjestelmällä on.
- Käyttäjävaa
timusten lukijat eivät ole ohjelmistotuotannon ammattilaisia

  - kuvaukset eivät saa sisältää ammattiterminologiaa tai tietotekniikka-jargonia

- Keskitytään käyttäjän kannalta oleellisiin vaatimuksiin, perustelut vaatimukselle syytä liittämään

## Esimerkki huonosta käyttäjävaatimuksesta

- CASE-työkalu suunnittelumallien editoimiseksi
  - 2.6 *Grid facilities* To assist in the positioning of entities on a diagram, the user may turn on a grid in either centimetres or inches, via an option on the control panel. Initially, the grid is off. The grid may be turned on and off at any time during an editing session and can be toggled between inches and centimetres at any time. A grid option will be provided on the reduce-to-fit view but the number of grid lines shown will be reduced to avoid filling the smaller diagram with grid lines." (Sommerville 2007)
- Käyttäjävaa
timukseksi liian yksityiskohtainen kuvaus- Osa alustustiedoista annettu (grid off), mutta kaikkea ei (tuumat vai sentit aluksi yksiköinä?)

## Kiinnitetty esitystapa

- **2.6.1 Grid facilities**
- **The editor shall provide a grid facility where a matrix of horizontal and vertical lines provide a background to the editor window.** This grid shall be a passive grid where the alignment of entities is the user's responsibility.
- *Rationale:* A grid helps the user to a tidy diagram with well-spaced entities. Although an active grid, where entities 'snap-to' grid lines can be useful, the positioning is imprecise. The user is the best person to decide where entities should be positioned.
- *Specification:* ECLIPSE/WS/Tools/DE/FS Section 5.6
- *Source:* Ray Wilson, Glasgow Office

- Sommerville (2007)

## Järjestelmävaatimukset

- Järjestelmävaatimukset määrittelevät yksityiskohtaisesti ja rakenteisella yhtenäisellä kuvaustekniikalla
  - järjestelmän tarjoamat palvelut
  - järjestelmän toiminnot
  - järjestelmän toiminnan rajoitteet
- Järjestelmävaatimukset määrittelevät täsmälleen, mitä järjestelmä tekee ja mitä ehtoja sille asetetaan

## Esimerkki: The LIBSYS system (Somerville 2007)

- Kirjastojärjestelmä joka tarjoaa käyttöliittymän, jonka kautta voi hakea artikkeleita useista tietokannoista eri kirjastoissa
- Käyttäjät voivat hakea, ladata ja tulostaa artikkeleita omaan käyttöönsä

## Vaatimusesimerkki (LIBSYS)

- LIBSYS shall keep track of all data required by copyright licensing agencies in the UK and elsewhere. (Käyttäjävaatimus)
- LIBSYS copyright licensing: (Järjestelmävaatimus)
  1. On making a request for a document from LIBSYS, the requestor shall be presented with a form that records details of the user and the request made.
  2. LIBSYS request forms shall be stored on the system for five years from the date of the request.
  3. All LIBSYS request forms must be indexed by user, by the name of the material requested and by the supplier of the request.
  4. LIBSYS shall maintain a log of all requests that have been made to the system.
  5. For material where authors' lending rights apply, loan details shall be sent monthly to copyright licensing agencies that have registered with LIBSYS.

## Luonnolliseen kieleen käyttöön liittyviä ongelmia

- Tarkkuus vs. dokumentin helppolukuisuus
- Useampia vaatimuksia voi sulautua yhteen
- Tulkinnan mahdollisuus: lukijoiden ja kirjoittajien tulisi ymmärtää ilmaisut samalla tavalla
- Samasta asiasta voidaan puhua useilla eri nimillä: sekaannuksen mahdollisuus
- Järjestelmävaatimuksia voidaan esittää täsmällisessä muodossa, esim. graafisia notaatioita käyttäen (UML) ja määrittelemällä standardimuotoisia lomakkeita

## Vaatimusten luokittelu

- Käyttäjä- ja järjestelmävaatimukset luokitellaan kolmeen ryhmään:
  1. *Toiminnalliset vaatimukset* (functional requirements)
  2. *Ei-toiminnalliset vaatimukset* (non-functional requirements)
  3. *Toimintaympäristövaatimukset* (domain requirements)

## Toiminnalliset vaatimukset

- Toiminnalliset vaatimukset annetaan käyttäjä- ja järjestelmävaatimuksina
- Toiminnalliset vaatimukset
  - Kuvaavat järjestelmän palvelut: mitä kaikkea järjestelmä tekee
  - Miten järjestelmä reagoi annettuihin syötteisiin
  - Miten järjestelmä toimii määritellyissä tilanteissa
  - Joskus myös, mitä järjestelmä ei missään tapauksessa tee

## Toiminnalliset vaatimukset: esimerkkejä (LIBSYS, Sommerville 2007)

1. "The user shall be able to search either all of the initial set of databases or select a subset from it."
  2. "The system shall provide appropriate viewers for the user to read documents in the document store."
  3. "Every order shall be allocated a unique identifier (ORDER\_ID) which the user shall be able to copy to the account's permanent storage area."
- Vaatimukset voivat olla hyvinkin erilaisilla tarkkuustasoilla (vrt. vaatimukset 1 ja 3)

© Juha Taina, Marko  
Salmenkivi ja Kjell  
Lemström, 2006-2011

581259 Ohjelmistotuotanto

103

## Vaatimusten epätasällisyys

- Epätasällisesti ilmaistut vaatimukset aiheuttavat usein ongelmia ja johtavat huonoon toteutukseen
- Käyttäjät ja ohjelmistokehittäjät saattavat tulkita eri tavoin
- Esim. edellisen kalvon vaatimuksissa käsite 'appropriate viewers'
  - Käyttäjän tarkoitus: eri formaateissa oleville dokumenteille räätälöidyt katselutoiminnot
  - Suunnittelija menee yli sieltä mistä aita on matalin ja tulkitsee, että yksi riittää

© Juha Taina, Marko  
Salmenkivi ja Kjell  
Lemström, 2006-2011

581259 Ohjelmistotuotanto

104

## Täydellisyys ja yhdenmukaisuus

- Toiminnallisten vaatimusten on oltava
  1. Täydellisiä: kaikki palvelut määritellään
  2. Yhdenmukaisia: määrittelyssä ei ole ristiriitaisuuksia.

© Juha Taina, Marko  
Salmenkivi ja Kjell  
Lemström, 2006-2011

581259 Ohjelmistotuotanto

105

## Ei-toiminnalliset vaatimukset

- Järjestelmän toiminnoille asetettavia rajoituksia ja reunaehtoja
- Ei-toiminnalliset vaatimukset koskevat koko järjestelmää.
  - Kaikki järjestelmän toiminnalliset vaatimukset toteuttavat järjestelmän ei-toiminnalliset vaatimukset

© Juha Taina, Marko  
Salmenkivi ja Kjell  
Lemström, 2006-2011

581259 Ohjelmistotuotanto

106

## Ei-toiminnalliset vaatimukset

- Ei-toiminnalliset vaatimukset ovat ainakin yhtä tärkeitä kuin toiminnalliset vaatimukset. Ne kertovat sen, miten käyttäjä kokee järjestelmän.
- Usein kriittisempiä kuin toiminnalliset vaatimukset (suorituskyky, luotettavuus)
- On mahdollista, että ei-toiminnallisen vaatimuksen pettäessä koko järjestelmä on käyttökelvoton.

© Juha Taina, Marko  
Salmenkivi ja Kjell  
Lemström, 2006-2011

581259 Ohjelmistotuotanto

107

## Ei-toiminnallisten vaatimusten tyypit

- Tuotteen käyttämiseen liittyvät vaatimukset
- Organisaation käytäntöihin liittyvät vaatimukset
- Järjestelmän ja organisaation ulkopuoliset vaatimukset
  - esim. kuinka järjestelmä on vuorovaikutuksessa muiden organisaatioiden järjestelmiin
  - lainsäädäntöön liittyvät vaatimukset
  - eettiset vaatimukset

© Juha Taina, Marko  
Salmenkivi ja Kjell  
Lemström, 2006-2011

581259 Ohjelmistotuotanto

108

### Esimerkkejä: Ei-toiminnalliset vaatimukset (LIBSYS, Sommerville 2007)

- Tuotteen käyttämiseen liittyvät vaatimukset  
8.1 The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.
- Organisaation käytäntöihin liittyvät vaatimukset  
9.3.2 The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.
- Järjestelmän ja organisaation ulkopuoliset vaatimukset  
7.6.5 The system shall not disclose any personal information about customers apart from their name and reference number to the operators of the system.

© Juha Taina, Marko Salmenkivi ja Kjell  
Lemström, 2006-2011 581259 Ohjelmistotuotanto 109

### Tavoitteet ja vaatimukset

- Ei-toiminnallisten vaatimusten määrittely täsmällisesti voi olla erityisen vaikeaa
- Epätasällisiä määrittelyjä voi olla hankala verifioida
- Tavoitteet vs. vaatimukset
  - Tavoitteita voivat olla esim. helppokäyttöisyys tai järjestelmän toimuminen virhetilanteista
- Ei-toiminnallinen vaatimus, joka on verifioitavissa
  - Tavoitteen operationalisointi: muuntaa mitattavissa olevaan muotoon
- Usein ei kuitenkaan onnistuta operationalisoimaan tavoitetta: tavoitteetkin hyödyllisiä, koska välittävät tietoa käyttäjän tarpeista

© Juha Taina, Marko Salmenkivi ja Kjell  
Lemström, 2006-2011 581259 Ohjelmistotuotanto 110

### Esimerkki: tavoite ja verifioitavissa oleva vaatimus (Sommerville 2007)

- **A system goal**
  - The system should be easy to use by experienced controllers and should be organised in such a way that user errors are minimised.
- **A verifiable non-functional requirement**
  - Experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

© Juha Taina, Marko Salmenkivi ja Kjell  
Lemström, 2006-2011 581259 Ohjelmistotuotanto 111

### (Toiminta)ympäristövaatimukset

- Sovellusalueen käsitteistä, käytännöistä ja lainalaisuuksista nousevia vaatimuksia järjestelmälle
- Ympäristövaatimukset liittyvät järjestelmän yhteistyön toimivuuteen sovellusympäristönsä kanssa
- Esimerkiksi järjestelmälle määriteltävät rajapinnat muihin järjestelmiin
- Voivat olla toiminnallisia tai ei-toiminnallisia
- Tärkeitä koska usein heijastelevat sovellusalueen perustavaa laatua olevia ominaisuuksia
- Jos vaatimukset eivät täyty, voi olla mahdotonta saada järjestelmää toimimaan tyydyttävästi

© Juha Taina, Marko Salmenkivi ja Kjell  
Lemström, 2006-2011 581259 Ohjelmistotuotanto 112

### Esimerkkejä ympäristövaatimuksista (LIBSYS, Sommerville 2007)

“There shall be a standard user interface to all databases which shall be based on the Z39.50 standard.”

“Because of copyright restrictions, some documents must be deleted immediately on arrival. Depending on the user's requirements, these documents will either be printed locally on the system server for manually forwarding to the user or routed to a network printer.”

© Juha Taina, Marko Salmenkivi ja Kjell  
Lemström, 2006-2011 581259 Ohjelmistotuotanto 113

### Esimerkki: junien automaattinen kulunvalvontajärjestelmä

“The deceleration of the train shall be computed as:

$$- D_{\text{train}} = D_{\text{control}} + D_{\text{gradient}}$$

where  $D_{\text{gradient}}$  is  $9.81\text{ms}^2$  \* compensated gradient/alpha and where the values of  $9.81\text{ms}^2/\alpha$  are known for different types of train.” (Sommerville 2007)

© Juha Taina, Marko Salmenkivi ja Kjell  
Lemström, 2006-2011 581259 Ohjelmistotuotanto 114

## Ympäristovaatimusten ongelmia

- Ymmärrettävyys
  - Vaatimukset ilmaistu sovellusalueen kielellä ja käsitteillä, jota ohjelmistokehittäjän voi olla vaikea ymmärtää
- Implisiittisyys
  - Sovellusalueen asiantuntijat tuntevat erikoisalansa niin hyvin, etteivät he ymmärrä muotoilla eksplisiittisesti kaikkia vaatimuksia

## Vaatimusdokumentti

- *Vaatimusdokumentti* (software requirements document tai software requirements specification, SRS) kuvaa yksiselitteisesti kaikki ohjelmistolle asetetut vaatimukset ja kehitetyt mallit.
- Vaatimusdokumentti *ei* ole suunnitteludokumentti: vaatimusdokumentin tehtävä on vastata kysymykseen: "Mitä järjestelmän pitää tehdä?"

## Vaatimusdokumentin rakenne

- Vaatimusdokumentin rakenteen pitää olla sellainen, että sitä voidaan käyttää
  - sopimuksena asiakkaan ja ohjelmistoyrityksen välillä,
  - suunnittelun lähtökohtana,
  - johdolle esiteltävänä yleisenä kuvauksena tehtävästä tuotteesta,
  - järjestelmätestauksen lähtökohtana jne.

## Vaatimusdokumentin sisältö

1. Esipuhe (preface)
  - Selvittää kenelle dokumentti on tarkoitettu
  - Kuvaa dokumentin versiohistorian
  - Yhteenveto edellisen version jälkeen tehdyistä muutoksista
2. Johdanto
  - Sisältää järjestelmän yleiskuvauksen, sen tärkeimmät tehtävät ja yhteistyön muiden järjestelmien kanssa
  - Voi sisältää myös lyhyen vanhan järjestelmän kuvauksen ja viitteet sen dokumentaatioon.

## Vaatimusdokumentin sisältö 2

3. Sanasto
  - Määrittelee dokumentissa käytetyt termit. Dokumentin tulee olla myös alan sanastoa tuntemattoman henkilön luettavissa.
4. Käyttjävaatimukset
  - Kuvaa toiminnalliset ja ei-toiminnalliset käyttäjävaatimukset luonnollisella kielellä ja kaavioilla. Myös mahdolliset käyttäjävaatimuksiin laskettavat ympäristövaatimukset luetellaan täällä.

## Vaatimusdokumentin sisältö 3

5. Järjestelmäarkkitehtuuri
  - Antaa yleiskuvan ohjelmiston rakenteesta
  - Näyttää palvelujen jakautumisen osajärjestelmiin ja komponentteihin
  - Sisältää myös tiedot jo valmiina olevista uudelleenkäytettävistä komponenteista
6. Järjestelmävaatimukset
  - Kuvaa yksityiskohtaiset toiminnalliset ja ei-toiminnalliset vaatimukset
  - Kuvaustapa on yhteinen kaikille järjestelmävaatimuksille

## Vaatimuskirjan sisältö 4

7. Järjestelmämallit
  - Sisältää yksityiskohtaisemmat mallit järjestelmän osajärjestelmistä, komponenteista ja niiden välisistä suhteista
  - Järjestelmämallit ovat suunnittelun perustana. Ne kuvataan yhdellä tai useammalla kuvaustekniikalla.
8. Järjestelmän elinkaari
  - Selittää järjestelmän käyttöaikana odotettavissa olevat laitteiston, käyttövaatimusten ja ohjelmiston vaatimusten muutokset

## Vaatimuskirjan sisältö 5

9. Liitteet
  - Sisältää sellaiset olemassaolevat dokumentit tai viitteet, jotka vaikuttavat tuotteeseen, mutta joita ei ole määritelty vaatimusmäärittelyssä.
  - Esimerkiksi käytettävän tietokannan hallintajärjestelmän kuvaus voi olla tällainen liitedokumentti.
10. Hakemisto