

## Ohjelmistojen vaatimusmäärittely syksy 2013/7

### Vaatimusten evoluutio

Harri Laine  
Helsingin yliopisto  
Tietojenkäsittelytieteen laitos

1

### Vaatimusten evoluutio

- Olipa vaatimusmäärittelyn eri vaiheet läpikäyty miten huolellisesti tahansa, vaatimusdokumenttia täytyy yleensä päivittää:
  - Havaitaan virheitä.
  - Sidosryhmien tavoitteet ja toiveet muuttuvat.
  - Sidosryhmien priorisoinnit muuttuvat.
  - Toimintaympäristö (liiketoiminta, muut järjestelmät) muuttuu.
  - Ongelmakentän lait tai oletukset muuttuvat.
- Muutoksia voi ilmaantua vaatimusmäärittelyprosessin aikana, ohjelmistokehityksen myöhemmissä vaiheissa ja järjestelmän käyttöönoton jälkeen.
- Muutokset aiheuttavat vaatimuksille *evoluutiota*: syntymistä, eriytymistä, valintaa, kuolemista ja muutoshistoriaa.

2

### Vaatimusten evoluutio

- Jokainen vaatimusmäärittelyn spiraalikierron tuottaa yhden *version* vaatimusdokumentista.
- Jokainen versio on joko *revisio* tai *variantti*:
  - Kyseessä on *revisio*, jos muutokset (korjaukset, parannukset) kohdistuvat yhden tietyn tuotteen vaatimuksiin.
  - Kyseessä on *variantti*, jos muutosten tuloksena saadaan vaatimusmäärittely samaan perusversioon perustuvalle mutta siitä selvästi eroavalle uudelle tuotteelle, jolla on omia ominaisuuksia ja joka on yleensä suunnattu jollekin erityiselle käyttäjäryhmälle.
- Jos tuotteella on useita varianteja, on kyseessä *tuoteperhe* (product family, product line).
- Kustakin revisiosta voi syntyä useita varianteja ja kustakin variantista useita revisioita.

3

### Vaatimusten evoluutio

- Esimerkkejä kokousjärjestelmän revisioista:
  1. Korjataan vaatimusmäärittelyn sisäisiä ristiriitoja.
  2. Otetaan osallistujien pyynnöstä mukaan uusi vaatimus, jolla kokouksen aika ja paikka saadaan tekstiviestinä kännykkään.
- Esimerkkejä kokousjärjestelmän varianteista:
  1. Kokouksesta ilmoitetaan sekä aika että paikka.
  2. Kokouksesta ilmoitetaan ainoastaan ajankohta (kokoukset pidetään aina samassa paikassa).
  3. Tärkeät osallistajat saavat kokouksen ajan ja paikan suoraan sähköiseen kalenteriinsa.

4

### Muutoksiin varautuminen

- Vaatimusten vakaus vaihtelee: stabiilit – epävakait
- Luokittelu stabiiliisuuden perusteella
  - Esim. pareittain vertailemalla kuten prioriteettien ja riskien kohdalla
  - Samaa muunnosyksikköön saman stabiiliusasteen vaatimuksia
- Erityisesti epävakaiden vaatimusten kohdalla kannattaisi kirjata perustelut, oletukset, alkuperä ja vaihtoehdot

5

### Muutoksiin varautuminen

- Ydinvaatimukset, joiden pitäisi toteutua kaikissa muunnelmissa - vakaita
- Tavoitteelliset ja käsitteelliset vakaampia kuin toimintatapaan liittyvät
- Keskeisiin tavoitteisiin liittyvät toiminnalliset vaatimukset vakaampia kuin ei toiminnalliset muutostarpeet
- Paljon vaihtoehtoja johtaa epävakaisiin vaatimuksiin
  - Ei tiedetä riittävästi
  - Tilanne muuttuu
- Ristiriitatilanteiden ratkaisut epävakaita
- Riskientorjunta ratkaisut epävakaita
- Vastuunjakoratkaisut epävakaita

6

### Vaatimusten jäljitettävyys

- Jotta vaatimusten evoluutiota voidaan hallita, pitäisi vaatimusten olla **jäljitettäviä** (traceable).
- Jäljitettävyys (traceability):
  - *mistä* vaatimus on peräisin
  - *miksi* se on tuotettu ja
  - *minne* se johtaa.

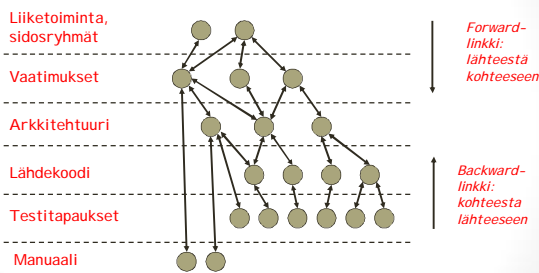
{ 7 }

### Vaatimusten jäljitettävyys

- Jäljitettävyys voidaan toteuttaa linkkiketjuna, josta löydetään:
    - Sidosryhmät ja liiketoiminnan tavoitteet, joista vaatimus on peräisin ( ← )
    - Ohjelmiston osat, joihin tietty vaatimus vaikuttaa ( → )
- Vaatimukset, joiden takia tietty ohjelmiston osa on tuotettu. ( ← )
- ← : taaksepäin jäljitettävyys, *backward traceability*  
→ : eteenpäin jäljitettävyys, *forward traceability*

{ 8 }

### Vaatimusten jäljitettävyys



{ 9 }

### Vaatimusten jäljitettävyys

- Myös samalla tasolla olevien määrittelyjen välillä voi olla riippuvuuksia, esimerkiksi
  - viittaus määritelmään
  - perustana oleva oletus
- Version sisäisiä riippuvuuksia
  - Yleinen riippuvuus  $A \rightarrow B = A:n$  muuttaminen voi edellyttää  $B:n$  muuttamista (*A vaikuttaa B:hen*, tai toisinpäin *B riippuu A:sta*)
- Erikoistapauksia:
  - $B \rightarrow$  uses  $A = A:n$  muuttaminen rikkoo  $B:n$  (B käyttää A:ta)
    - Esimerkiksi toimintakuvaus perustuu käsitelmämäärittelyyn, määrittelyn muuttaminen johtaa virheelliseen toimintaan
  - $B \rightarrow$  derived from  $A = B$  on johdettu A:sta, A on B:n lähtökohta
    - Esim. A on tavoite ja B joukko järjestelmävaatimuksia sen saavuttamiseksi
- Jäljitettävyyslinkitystä voidaan hyödyntää mm. testauksessa ("mitä vaatimuksia vasten on tämä komponentti testattava") ja ylläpidossa ("mihin ohjelmiston osiin tämän vaatimuksen muuttaminen vaikuttaa").

{ 10 }

### Vaatimusten jäljitettävyys

- Riippuvuussuhteen yhtenä ongelmana on karkeusaste (granularity)
  - Hyvin hienojakoisesti kuvatut riippuvuudet johtavat suureen määrään riippuvuuksia
  - Hienojakoisuus on paikallaan
    - Kriittisten piirteiden kohdalla (esim. turvallisuus)
    - Epävakaiden piirteiden kohdalla
- Riippuvuusmatriisit, hyvä tekniikka riippuvuuksien kirjaamiseen.
  - Kaikki kohteet samassa matriisissa tai
  - Kohderyhmien väliset matriisit esim. tavoite x järjestelmävaatimus

{ 11 }

### Muutostenhallinta

- Muutostarpeita syntyy eri vaiheissa ohjelmiston elinkaarta
- Muutostoiveista muutostarpeiksi
  - Mihin tavoitteisiin, vaatimuksiin, oletuksiin muutos kohdistuu
  - Missä tilanteessa tarve on tullut esiin
  - Perustelu (seuraukset)
  - Muutosta toivonut sidosryhmä
  - Kiireellisyys
  - Arvio vaikutuksista
  - Arvio kustannuksista ja tarvittavista resursseista

{ 12 }

### Muutostenhallinta

- Arviointielin (review board) arvioi muutostarpeet ja päättää tehtävistä muutoksista
  - Arviointielin erilainen tilanteesta riippuen
  - Päätösvalta
- Tehtävät
  - Selvittää mihin yhteyteen muutostarve liittyy (jäljitettävyys)
  - Tarkastaa ovatko esitetyt muutokset perusteltuja
  - Arvioida muutoksen hyödyt suhteessa tavoitteisiin
  - Arvioida muutoksen vaikutukset muihin kohteisiin (eteenpäin jäljitettävyys)
  - Paikantaa mahdolliset muutosehdotukseen sisältyvät ristiriidat
  - Arvioida muutoksen ja muuttamatta jättämisen riskit
  - Arvioida ehdotetun muutoksen kannattavuus ja kustannukset
  - Laittaa muutokset tärkeysjärjestykseen ja päättää poisjätettävistä

13

### Vaatimusten ja oletusten tarkkailu

- Vaatimusten määrittelyn yhteydessä esitetään arvioita ja oletuksia toimintaympäristöstä ja ohjelmiston käytöstä.
- Kun ohjelmisto otetaan käyttöön toimintaympäristö muuttuu ja oletukset eivät välttämättä pidä paikkaansa
- Vaatimusten tarkkailulla (requirements monitoring) pyritään löytämään näitä ongelmia

14

### Vaatimusten ja oletusten tarkkailu

- Vaatimusanalyysin aikana määritellään vaatimukset, joita pitäisi tarkkailla
- Toteutettavaan ohjelmistoon määritellään tarkkailumonitorit tutkittavia tapauksia varten
- Toteutetaan tarkkailuvälineistö osana ohjelmistoa (mahdollisesti pois päältä kytkettävänä)
- Käytetään välineistöä suoritusaikana. Välineistö voi tuottaa lokia, varoituksia, tilastoja, mutta myös ohjata vaihtoehtoisin toimintatapoihin.

15