

#### 4. Vaatimusmäärittely

- Laadukkaiden ohjelmistojen tuottaminen ei ole helppo tehtävä.
  - Jos se olisi helppoa, kaikki tekisivät laadukkaita ja *edullisia* ohjelmia.
- Sen lisäksi, että ohjelman täytyy toimia virheettömästi, sen täytyy täyttää sille asetetut implisiittiset ja eksplisiittiset vaatimukset.

Kevät 2005

Ohjelmistotuotanto / Taina

1

#### Implisiittiset vaatimukset

- Implisiittiset vaatimukset ovat ominaisuuksia, joiden oletetaan olevan kaikilla kehitettävän tyyppisillä laadukkailla ohjelmistoilla.
  - Tällaisia ovat esimerkiksi ylläpidettävyys, yhdenmukainen käyttö ja virheettömyys.
- Implisiittisiä vaatimuksia ei välttämättä listata erikseen, mutta tuotettavan ohjelmiston tulee toteuttaa ne.

Kevät 2005

Ohjelmistotuotanto / Taina

2

#### Eksplisiittiset vaatimukset

- Eksplisiittiset vaatimukset ovat vaatimuksia, jotka on lueteltu erikseen ja jotka tuotettavan ohjelman täytyy toteuttaa. Nämä selvitetään vaatimusmäärittelyssä.
  - Esimerkiksi kaikki ohjelman toiminnot ovat eksplisiittisiä vaatimuksia. Samoin vasteaikoihin ja käyttäjämääriin liittyvät vaatimukset ovat eksplisiittisiä.

Kevät 2005

Ohjelmistotuotanto / Taina

3

#### Vaatimusmäärittelyn tavoitteet

- Vaatimusmäärittelyn tavoitteena on selvittää ohjelmistolle asetettavat vaatimukset sellaisella tarkkuudella, että niiden perusteella voidaan tuottaa haluttu ohjelmisto.
- Vaatimukset ovat eritasoisia. Osa on hyvin korkealla tasolla, osa menee syvälle yksityiskohtiin.

Kevät 2005

Ohjelmistotuotanto / Taina

4

#### Vaatimusten luonne

- Vaatimusten on oltava
  - virheettömiä. Vaatimusmäärittelyssä tehty virhe heijastuu kaikkiin prosessin seuraaviin vaiheisiin ja tulee kalliiksi.
  - ristiriidattomia. Ristiriitaisuutta voidaan vähentää huolellisella vaatimusten analysoinnilla.
  - täydellisiä. Käytännössä tämä ei toteudu, sillä vaatimukset muuttuvat projektin aikana.

Kevät 2005

Ohjelmistotuotanto / Taina

5

#### Vaatimusten luonne II

- realistisia. Vaatimusmäärittelyssä selvitetään myös järjestelmäsuunnittelun ja toteutusympäristön asettamat rajoitukset.
- tarpeellisia. Tarpeellisten vaatimusten selvittäminen on yllättävän vaikea tehtävä. Vaatimukset tulee priorisoida.
- todennettavissa. Jokainen vaatimus on löydettävä tuotteesta, ja tuotteen jokaisen ei-triviaalin ominaisuuden on oltava tunnistettu vaatimus.

Kevät 2005

Ohjelmistotuotanto / Taina

6

### Vaatimusten luonne III

- jäljitettävissä. Jokaisella vaatimuksella on esittäjä, joka on voitava selvittää tarpeen vaatiessa.
- yksiselitteisiä. Moniselitteisyys syntyy puhutun ja kirjoitetun kielen moniselitteisyydestä. Tämän johdosta vaatimusmäärittelyssä pitää aina tehdä vaatimusten perusteella malli, jota voidaan käyttää vaatimusten validointiin.

Kevät 2005

Ohjelmistotuotanto / Taina

7

### 4.1. Vaatimustyypit

- Vaatimuksia käytetään eri tarkoituksiin:
  - Ne kuvaavat tehtävää ohjelmistoa.
  - Ne toimivat sopimuksena asiakkaan ja ohjelmistoyrityksen välillä.
  - Ne tarkentavat ja selventävät ohjelmiston toiminnan yksityiskohtia.
  - Niitä käytetään suunnittelun perustana.
- Eri tarkoitukset vaativat eri tyyppisiä kuvauksia vaatimuksista.

Kevät 2005

Ohjelmistotuotanto / Taina

8

### Vaatimusten jaottelu

- Vaatimusten erilaisten tarkoitusten johdosta Sommerville jakaa vaatimukset kolmeen tyyppiin:
  - käyttäjävaatimukset (user requirements),
  - järjestelmävaatimukset (system requirements) ja
  - ohjelmiston määrittelykuvaukset (software design specification).

Kevät 2005

Ohjelmistotuotanto / Taina

9

### Käyttäjävaatimukset

- Käyttäjävaatimukset listaavat ohjelmiston tarjoamat palvelut ja ohjelmistolle asetettavat rajoitukset.
- Käyttäjävaatimukset kuvataan luonnollisella kielellä ja kaavioilla. Käyttäjän vaatimuksia käytetään analyysin pohjana ja yhteydenpidossa asiakkaan kanssa.

Kevät 2005

Ohjelmistotuotanto / Taina

10

### Järjestelmävaatimukset

- Vaatimukset kuvaavat ohjelmiston palvelut ja rajoitukset yksityiskohtaisesti.
- Kaikki vaatimukset kuvataan samalla muodollisella formaatilla.
- Järjestelmävaatimuksia käytetään sopimuksena asiakkaan ja ohjelmistoyrityksen välillä.

Kevät 2005

Ohjelmistotuotanto / Taina

11

### Määrittelykuvaukset

- Määrittelykuvaus on kattava kuvaus ohjelmistosta.
- Kuvauksessa käytetään standardoituja menetelmiä, kaavioita ja strukturoitua esitystapaa.
- Kuvaus toimii suunnittelun esiasteena.
- Kuvauksessa ei mennä yksityiskohtiin, vaan pysytään korkealla abstraktiolla.

Kevät 2005

Ohjelmistotuotanto / Taina

12

## 4.2. Vaatimusten luokittelu

- Vaatimukset voidaan luokitella toiminnallisiin (functional) ja ei-toiminnallisiin (non-functional) vaatimuksiin.
  - Toiminnalliset vaatimukset määrittelevät, mitä palveluja ohjelmisto tarjoaa.
  - Ei-toiminnalliset vaatimukset määrittelevät toimintojen järjestelmälle asettamat reunaehdot.

Kevät 2005

Ohjelmistotuotanto / Taina

13

## Toiminnalliset vaatimukset

- Toiminnalliset vaatimukset määrittelevät, mitä palveluja ohjelmiston on tarjottava, miten ohjelmisto reagoi syötteisiin ja miten se käyttäytyy annetuissa tilanteissa.
- Joskus toiminnalliset vaatimukset myös määrittelevät, mitä ohjelmiston ei pidä tehdä.

Kevät 2005

Ohjelmistotuotanto / Taina

14

## Toiminnallisten vaatimusten piirteitä

- Toiminnalliset vaatimukset ovat intuitiivisesti selkeitä. Tarkoitus on listata, mitä palveluja järjestelmä tarjoaa.
- Toiminnallisiin vaatimuksiin kuuluvat luonnollisesti ohjelmiston normaalit palvelut, mutta niihin kuuluvat myös virhetilanne- ja poikkeuskäsittelyn tarjoamat toiminnot ja toipumistavat.

Kevät 2005

Ohjelmistotuotanto / Taina

15

## Toiminnallisten vaatimusten piirteitä II

- Toiminnalliset vaatimukset etsitään alkaen suurista kokonaisuuksista ja edeten kohti pienempiä osia.
- Vaatimukset pyritään kartoittamaan eri näkökulmista, jotta vaatimuksista saadaan mahdollisimman kattavat.

Kevät 2005

Ohjelmistotuotanto / Taina

16

## Ei-toiminnalliset vaatimukset

- Ei-toiminnalliset vaatimukset määrittelevät rajoitukset ja reunaehdot toiminnallisille vaatimuksille.
- Ei-toiminnalliset vaatimukset eivät liity suoraan palveluihin, vaan kertovat, mitä ehtoja järjestelmän on täytettävä, jotta toiminnalliset vaatimukset voidaan toteuttaa.

Kevät 2005

Ohjelmistotuotanto / Taina

17

## Ei-toiminnallisten vaatimusten piirteitä

- Koska ei-toiminnalliset vaatimukset eivät liity suoraan palveluihin, ne eivät ole yhtä intuitiivisia kuin toiminnalliset vaatimukset.
- Ei-toiminnalliset vaatimukset asettavat ohjelmistolle ja sen toteutukselle rajoituksia.
  - Rajoitus: jokin sääntö, joka ohjelmiston on toteutettava.

Kevät 2005

Ohjelmistotuotanto / Taina

18

## Ei-toiminnallisten vaatimusten piirteitä II

- Ei-toiminnallisten vaatimusten löytäminen, luokittelu ja validointi on vaikeampaa kuin toiminnallisten vaatimusten, sillä ne on johdettava käyttäjä- ja järjestelmävaatimuksista.
- Apuna voidaan käyttää luetteloa tyyppillisistä tehtävän ohjelmiston tyyppiä olevissa tuotteissa olevista ei-toiminnallisista vaatimuksista.

Kevät 2005

Ohjelmistotuotanto / Taina

19

## Ei-toiminnallisten vaatimusten luokkia

- Seuraava luokittelu voi auttaa löytämään ei-toiminnallisia vaatimuksia:
  - Nopeus: tapahtumia sekunnissa, vasteajat, päivitysajat ym.
  - Koko: ohjelmiston koko, syöttö-/tulostiedoston koko ym.
  - Käytettävyys: oppimiskäyrä, opastus ym.
  - Luotettavuus: vikasietoisuusaste, saavutettavuusaika ym.

Kevät 2005

Ohjelmistotuotanto / Taina

20

## Ei-toiminnallisten vaatimusten luokkia

- Sitkeys (robustness): toipumisaika, kaatumistodennäköisyys, virhetilanteessa tietojen katoamistodennäköisyys ym.
- Siirrettävyys (portability): alustariippuvan koodin määrä, tuettavien järjestelmäalustojen määrä ym.
- Ei-toiminnalliset vaatimukset pyritään esittämään täsmällisillä lukuarvoilla.

Kevät 2005

Ohjelmistotuotanto / Taina

21

## Luokitteluesimerkkejä

- Esimerkiksi vaikka näin:
  - Ohjelmistoon pitää pystyä tekemään kyselyjä kulloinkin paikalla olevista käyttäjistä (toiminnallinen vaatimus).
  - Ohjelmiston on selvittävä 25 kyselystä sekunnissa (ei-toiminnallinen vaatimus).
  - Ohjelmiston on reagoitava hälytykseen hälytysmerkkiäänellä ja vilkkuvalle näytöllä (toiminnallinen vaatimus).

Kevät 2005

Ohjelmistotuotanto / Taina

22

## Luokitteluesimerkkejä II

- Hälytykseen on reagoitava alle 0,1 sekunnin sisällä hälytyksen aiheuttaneen signaalin saapumisesta (ei-toiminnallinen vaatimus).
- Ohjelmistossa on oltava ylläpidolle tarjotut palvelut toiseen käyttöympäristöön siirtymiseksi (toiminnallinen vaatimus).
- Ohjelmiston on toimittava Windows- ja Linux-ympäristöissä (ei-toiminnallinen vaatimus).

Kevät 2005

Ohjelmistotuotanto / Taina

23

## Ympäristövaatimukset

- Ympäristövaatimukset (domain requirements) ovat vaatimuksia, joiden lähtökohtana on ohjelmistoa ympäröivä toimintaympäristö käyttäjien sijaan.
- Ympäristövaatimukset voivat olla joko toiminnallisia tai ei-toiminnallisia vaatimuksia.

Kevät 2005

Ohjelmistotuotanto / Taina

24

### Ympäristövaatimusten luonne

- Ympäristövaatimukset ovat tärkeitä, sillä ne määrittelevät ehdot, jotka ohjelmiston tulee täyttää, jotta se voi toimia järjestelmässä.
- Ympäristövaatimuksia ovat esimerkiksi ohjelmistolle määriteltävät rajapinnat muihin ohjelmistokomponentteihin.

Kevät 2005

Ohjelmistotuotanto / Taina

25

### Yhteenveto vaatimuksista

- Vaatimus voi olla:
  - Implisiittinen
    - Ei-toiminnallinen
      - Järjestelmävaatimus
      - Ympäristövaatimus
  - Eksplisiittinen
    - Toiminnallinen/Ei-toiminnallinen
      - Käyttäjävaatimus
      - Järjestelmävaatimus
      - Ympäristövaatimus

Kevät 2005

Ohjelmistotuotanto / Taina

26

### 4.3. Vaatimusmäärittelyprosessin vaiheet

- Vaatimusmäärittelyn työvaiheet ovat:
  - Kelpoisuus selvitys (feasibility study).
    - Katsotaan, kannattaako ohjelmistoa ylipäänsä toteuttaa.
  - Vaatimusten kartoitus ja analyysi (requirements elicitation and analysis).
    - Kerätään vaatimuksia asiakkaan kanssa.
    - Mallinnetaan löydettyjen vaatimusten perusteella kehitettävää järjestelmää.
    - Johdetaan malleista uusia vaatimuksia.

Kevät 2005

Ohjelmistotuotanto / Taina

27

### Vaatimusmäärittelyprosessin vaiheet II

- Vaatimusten täsmennys (requirements specification).
  - Jaetaan kartoitetut vaatimukset käyttäjävaatimuksiin ja järjestelmävaatimuksiin.
  - Kirjataan kartoitetut vaatimukset dokumentiksi.
- Vaatimusten validointi (requirements validation).
  - Varmennetaan, että löydetty vaatimukset todella määrittelevät asiakkaan haluaman järjestelmän.

Kevät 2005

Ohjelmistotuotanto / Taina

28

### Kelpoisuus selvitys

- Kelpoisuus selvitys on lyhyt esivaihe vaatimusmäärittelylle. Siinä katsotaan
  1. tuoko kehitettävä järjestelmä lisäarvoa asiakkaalle.
  2. voidaanko järjestelmä toteuttaa nykyisellä teknologialla, projektille varatulla aikataululla ja budjetilla.
  3. voidaanko järjestelmä integroida jo olemassaoleviin järjestelmiin.

Kevät 2005

Ohjelmistotuotanto / Taina

29

### Kelpoisuus selvitys II

- Lisäksi, vaikka Sommerville ei sitä listaa:
  4. kannattaako järjestelmä toteuttaa, vai voidaanko vastaava järjestelmä ostaa valmiina. Onko ostettava järjestelmä sovittavissa asiakkaan tarpeisiin?
- Tuloksena saadaan päätös siitä, kannattaako järjestelmän kehitystyötä jatkaa.

Kevät 2005

Ohjelmistotuotanto / Taina

30

## Vaatimusten kartoitus ja analyysi

- Tässä työvaiheessa projektiryhmä työskentelee yhteistyössä asiakkaiden ja järjestelmän loppukäyttäjien kanssa selvittääkseen
  - missä järjestelmää tullaan käyttämään,
  - mitä palveluita järjestelmän pitää tarjota,
  - mitä ei-toiminnallisia vaatimuksia ja rajoituksia järjestelmällä on.

Kevät 2005

Ohjelmistotuotanto / Taina

31

## Sidosryhmät

- Vaatimusmäärittely koskettaa useaa *sidosryhmää*.
  - Sidosryhmä = henkilö tai ryhmä, joka on suoraan tai välillisesti tekemisissä kehitettävän järjestelmän kanssa.
- Eri sidosryhmillä on erilaisia tarpeita kehitettävälle järjestelmälle: ristiriitoja.
- Sidosryhmät eivät välttämättä osaa kertoa, mitä kaipaavat järjestelmältä.

Kevät 2005

Ohjelmistotuotanto / Taina

32

## Vaatimusten kartoituksen ja analyysin osavaiheet

1. Vaatimusten löytäminen (requirements discovery).
  - Etsitään vaatimuksia sidosryhmien edustajien kanssa.
2. Vaatimusten luokittelu ja ryhmittely (requirements classification and organisation).
  - Kootaan vaatimukset ryhmiin, missä kussakin ryhmässä on toisistaan riippuvia vaatimuksia.

Kevät 2005

Ohjelmistotuotanto / Taina

33

## Vaatimusten kartoituksen ja analyysin osavaiheet II

3. Vaatimusten priorisointi ja vaatimuksista neuvottelu (requirements prioritisation and negotiation).
    - Priorisoidaan vaatimukset ja ratkotaan sidosryhmien vaatimuksien väliset ristiriidat.
  4. Vaatimusten dokumentointi (requirements documentation).
    - Dokumentoidaan löydetty vaatimukset.
- Osavaiheita toistetaan, kunnes uusia vaatimuksia ei enää löydetä.

Kevät 2005

Ohjelmistotuotanto / Taina

34

## Vaatimusten täsmennys

- Työvaiheessa
  - vaatimukset jaetaan käyttäjä- ja järjestelmävaatimuksiin,
  - tarkistetaan, että tehdyt määrittelykuvaukset vastaavat vaatimuksia.
- Tuloksena saadaan lopulliset vaatimuslistat ja määrittelykuvaukset.
- Jos kuvaukset eivät ole riittävät, palataan edellisiin osavaiheisiin.

Kevät 2005

Ohjelmistotuotanto / Taina

35

## Vaatimusten validointi

- Viimeisenä vaiheena edellisissä työvaiheissa jalostetut vaatimukset validoidaan.
- Tarkoituksena on selvittää, että saatu vaatimuslista ja vaatimusten perusteella kehitetyt mallit ovat yhtenäiset ja toteuttavat vaatimuksille asetetut ominaisuudet (katso "Vaatimusten luonne").

Kevät 2005

Ohjelmistotuotanto / Taina

36

## Skenaariot

- Eräs tapa kartoittaa vaatimuksia on pyytää asiakkaalta skenaarioita (scenarios) nykyisestä ja tulevasta järjestelmän toiminnasta.
- Skenaariot ovat kuvauksia järjestelmän käytöstä eri näkökulmista.
- Hyvin tehdyt skenaariot antavat hyvän kuvan järjestelmän toiminnasta.

Kevät 2005

Ohjelmistotuotanto / Taina

37

## Skenaarion sisältö

- Skenaario sisältää yleensä
  - järjestelmän alkutilan,
  - normaalin toiminnan kuvauksen,
  - mahdollisesti poikkeustilanteiden kuvauksen,
  - tiedot muista samanaikaisista skenaarioon vaikuttavista tapahtumista ja
  - järjestelmän tilan kuvauksen toiminnan päättymisen jälkeen.

Kevät 2005

Ohjelmistotuotanto / Taina

38

## Skenaarioiden kuvaus

- Skenaariot voidaan kuvata yhdessä tai erikseen.
- Jos ne kuvataan yhdessä, ensin kuvataan yksi onnistunut skenaario ja sitten siihen liittyvät poikkeustapaukset.
- Erikseen kuvattaessa skenaarioiden välinen yhteys selviää siitä käyttötapauksesta, mitä ne kuvaavat.

Kevät 2005

Ohjelmistotuotanto / Taina

39

## Skenaarioesimerkki

- Skenaario "Talleta nimetty tiedosto":
  - Alkutila: käyttäjä on muokannut vanhaa tiedostoa.
  - Onnistunut skenaario:
    1. Valitse valikosta "Tallenna..." tai paina <Ctrl>+S
    2. Ohjelma näyttää dialogin "Tiedosto on jo olemassa. Kirjoitetaanko silti (Kyllä/Ei)"
    3. Valitse dialogista "Kyllä"
    4. Ohjelma kirjoittaa muutokset levyille.

Kevät 2005

Ohjelmistotuotanto / Taina

40

## Skenaarioesimerkki II

- Lopputila: Tiedosto on talletettu. Ohjelma tietää, että muokattava tiedosto vastaa levyllä olevaa tiedostoa.
- Vaihtoehto 3a: valinta ei (laajennos kohdasta 3 eteenpäin):
  - 3a1. Valitse dialogista "Ei"
  - Lopputila 3a: Tiedostoa ei ole talletettu. Ohjelma tietää, että muokattava tiedosto ja levyllä oleva tiedosto eivät ole yhtenevät.

Kevät 2005

Ohjelmistotuotanto / Taina

41

## Skenaarioesimerkki - III

- Vaihtoehto 4a: talletus ei onnistu (laajennos kohdasta 4 eteenpäin):
  - 4a1. Ohjelma yrittää tallettaa tiedoston levyille
  - 4a2. Tallennus päättyy tiedostovirheeseen
  - 4a3. Ohjelma palauttaa dialogin "Tiedoston tallennus ei onnistunut tiedostovirheen johdosta. Paina OK"
  - 4a4. Valitse "OK"
  - Lopputila: Kuten Lopputila 3a

Kevät 2005

Ohjelmistotuotanto / Taina

42

### Skenaarioiden tarkkuus

- Edellinen esimerkki sisälsi oikeastaan kolme skenaariota.
  - Jokainen kuvauksen skenaario olisi voitu kuvata alusta asti yksin, mutta yhdistämällä skenaarioiden välille saatiin selkeä yhteys.
- Skenaariot kannattaa kuvata sellaisella tarkkuudella, että toiminnan vaiheet tulevat selville, mutta toteutuksen yksityiskohdat jäävät pois.

Kevät 2005

Ohjelmistotuotanto / Taina

43

### Käyttötapaukset

- Käyttötapaus (use-case) sisältää yhden tai useamman järjestelmän skenaarion.
- Samankaltaiset tai samoihin järjestelmän osiin vaikuttavat skenaariot kootaan saman käyttötapauksen alle.
- Skenaarioiden jaottelu käyttötapauksiin on suunnittelupäätös.

Kevät 2005

Ohjelmistotuotanto / Taina

44

### Käyttötapauksen tehtävä

- Jokainen käyttötapaus kertoo yhden tavan toimia kehitettävässä järjestelmässä, toiminnan tehtävät ja toimintaan osallistuvat sidosryhmät.
- Käyttötapaukset voivat olla selväkielistä tekstiä tai niihin voidaan käyttää sopivaa kuvaustekniikkaa.

Kevät 2005

Ohjelmistotuotanto / Taina

45

### Käyttötapauksen määritelmä

- Käyttötapaus (use case) on yksi eniten kuormitettuja termejä. Käyttötapaus voi kirjoittajasta riippuen olla
  - joukko toisiinsa liittyviä skenaarioita,
  - palvelu ja siihen liittyvät sidosryhmät,
  - jokainen skenaario yksinään,
  - toiminnan tavoitteen kuvaus jne.
- Kurssilla käytämme ensimmäistä määritelmää.

Kevät 2005

Ohjelmistotuotanto / Taina

46

### Esimerkkejä käyttötapauksista

- Seuraavat voisivat olla tekstintoittoimen käyttötapauksia:
  - "Lihavoi teksti" - skenaariot sisältävät eri tavat lihavoida tekstiä.
  - "Tee sisällysluettelo" - sisällysluettelo voidaan rakentaa usealla eri tavalla.
  - "Talleta tiedosto" - talletustapa vaihtelee sen mukaan onko tiedostolla nimi ja onko tiedosto jo ennestään olemassa.

Kevät 2005

Ohjelmistotuotanto / Taina

47

### Käyttötapausyhteenveto

- Eli käyttötapaus
  - kuvaa jonkin käyttäjälle näkyvän toiminnon,
  - voi olla suppea tai laaja,
  - perustuu tunnettuun alkutilaan (toiminto käynnistyy) ja
  - tuottaa jonkin tuloksen (toiminto valmistuu).

Kevät 2005

Ohjelmistotuotanto / Taina

48



### Mihin käyttötapaukset kelpaavat?

- Käyttötapaukset ovat erinomainen tapa kuvata sekä vanhan että uuden järjestelmän toimintaa.
- Samoin käyttötapaukset sopivat vaatimusten kartoitukseen. Vaikka käyttötapaus ei ole vaatimus, sen avulla voidaan generoida sekä käyttäjä- että järjestelmävaatimuksia.

Kevät 2005

Ohjelmistotuotanto / Taina

49

### Vielä käyttötapauksista

- Käyttötapauksia voidaan käyttää mallinnuksen apuvälineenä **ja** mallinnusta voidaan käyttää keinona löytää uusia käyttötapauksia.
- Käyttötapaukset eivät kerro mitään ohjelman sisäisestä rakenteesta, kuten luokista.

Kevät 2005

Ohjelmistotuotanto / Taina

50

### Skenaariot ja käyttötapaukset

- Skenaarion ja käyttötapauksen ero on häilyvä. Yhden henkilön käyttötapaus on toisen skenaario. Skenaarioiden ryhmittely on aina suunnittelupäätös.
- Skenaarioita ja käyttötapauksia voidaan tarkentaa käymällä asiakkaan luona katsomassa nykyistä toimintaa.

Kevät 2005

Ohjelmistotuotanto / Taina

51

### 4.4 Vaatimusmäärittelyn hallinta

- Vaatimusmäärittely on prosessi, joten se tarvitsee hallintaa.
- Hallinnan tehtävä on organisoida työvaiheet ja varustautua siihen, että vaatimukset muuttuvat projektin edetessä.
- Myös vaatimusmäärittelyyn kuuluu riskienhallintaa. Riskit tunnistetaan ja niihin varaudutaan.

Kevät 2005

Ohjelmistotuotanto / Taina

52

### 4.5. Vaatimusdokumentti

- Vaatimusmäärittelyn lopputuloksena saadaan vaatimusdokumentti.
- Se kuvaa yksiselitteisesti kaikki ohjelmistolle asetetut vaatimukset ja kehitetyt mallit.
- Dokumentin rakenne riippuu tuotettavan ohjelman tyypistä.
  - Esim. tietojärjestelmän dokumentti eroaa upotetun järjestelmän dokumentista.

Kevät 2005

Ohjelmistotuotanto / Taina

53

### Vaatimusdokumentille asetetut vaatimukset

- Vaatimusdokumentti
  - määrittelee ulkoisen toiminnan: mitä ohjelmisto tarjoaa.
  - määrittelee toteutuksen rajoitukset.
  - on helposti päivitettävissä.
  - toimii viitteenä ohjelmiston ylläpitäjille.
  - sisältää arvion tuotteen elinkaaresta.
  - käsittelee myös virhetilanteet.

Kevät 2005

Ohjelmistotuotanto / Taina

54

## Vaatimusdokumentin rakenne

- Vaatimusdokumentti hyväksytään sekä asiakkaan että ohjelmistoyrityksen puolelta. Sen pitää olla rakenteeltaan sellainen, että kaikki sidosryhmät voivat käyttää sitä:
  - asiakkaat, johtoryhmä, laadunvalvontaryhmä, halinto, projektipäällikkö, projektiryhmä ym.

Kevät 2005

Ohjelmistotuotanto / Taina

55

## Vaatimusdokumentin sisältö

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

Kevät 2005

Ohjelmistotuotanto / Taina

56

## Vaatimusdokumentin sisältö - II

3. Sanasto
  - Määrittelee dokumentissa käytetyt termit. Dokumentin tulee olla myös alan sanastoa tuntemattoman henkilön luettavissa.
4. Käyttäjän vaatimukset
  - Määrittelee käyttäjän vaatimukset luonnollisella kielellä ja kaavioilla. Myös mahdolliset käyttäjän vaatimuksiin laskettavat ympäristövaatimukset luetellaan täällä.

Kevät 2005

Ohjelmistotuotanto / Taina

57

## Vaatimusdokumentin sisältö - III

5. Järjestelmävaatimukset
  - Määrittelee yksityiskohtaiset toiminnalliset ja ei-toiminnalliset vaatimukset.
6. Ympäristövaatimukset
  - Määrittelee ympäristön ohjelmistolle asettamat rajoitukset ja vaatimukset.
7. Järjestelmäarkkitehtuuri
  - Antaa yleiskuvan ohjelmiston rakenteesta.
  - Luku sisältää myös tiedot jo valmiina olevista ohjelmistokomponenteista.

Kevät 2005

Ohjelmistotuotanto / Taina

58

## Vaatimusdokumentin sisältö - IV

8. Järjestelmämallit
  - Sisältää yksityiskohtaisemmat mallit järjestelmän osajärjestelmistä, komponenteista ja niiden välisistä suhteista.
  - Järjestelmämallit ovat suunnittelun perustana. Ne kuvataan yhdellä tai useammalla kuvaustekniikalla.
9. Järjestelmän kehitys
  - Selittää odotetut laitteiston, käytön ja ohjelmiston vaatimusten muutokset.

Kevät 2005

Ohjelmistotuotanto / Taina

59

## Vaatimusdokumentin sisältö - V

10. 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.
11. Hakemisto

Kevät 2005

Ohjelmistotuotanto / Taina

60