

- Requirements elicitation
- Requirements specification
- Requirements validation
- Requirements change

# Vaatimusten spesifiointi

# Vaatimusten määrittely (specification)

- Kirjoitetaan käyttäjä- ja järjestelmävaatimukset vaatimusten määrittelydokumenttiin.
- Myös sellaisten loppukäyttäjien ja asiakkaiden pitää ymmärtää käyttäjävaatimukset, joilla ei ole teknistä koulutusta.
- Järjestelmävaatimukset ovat yksityiskohtaisempia ja voivat sisältää enemmänkin teknistä tietoa.
- Vaatimukset voivat olla osa toimitussopimusta
  - Siksi niiden pitää olla niin täydellisiä kuin mahdollista.

# Järjestelmävaatimusten määritelmien kirjoitustapoja

- Requirements elicitation
- Requirements specification
- Requirements validation
- Requirements change

Notaatio	Kuvaus
Luonnollinen kieli	Vaatimukset listataan numeroituina virkkeinä. Jokainen virke ( <a href="#">sentence</a> ) esittää yhden vaatimuksen.
Strukturoitu luonnollinen kieli	Vaatimukset kirjoitetaan määrättyä standardisapluunaa (template) käyttäen.
Suunnittelun (design) kuvauskieli	Käytetään <a href="#">ohjelmointikielten kaltaista kieltä</a> . Tällaiset ovat nykyään harvinaisia, mutta silti sopivat käyttöliittymän määritelmiin.
Graafinen notaatio	Graafisia kuvauskieliä niiden lisädokumentaation kanssa käytetään toiminnallisten vaatimusten kuvaamiseen. <a href="#">UML:n käyttötapaus- (use case)</a> ja <a href="#">sekvenssikaaviot</a> ovat erittäin yleisiä
Matemaattiset spesifikaatiot	Perustuvat matemaattisiin konsepteihin kuten äärellisiin tilakoneisiin tai joukkoihin.  Yleensä käyttäjät eivät halua niitä käytettävän, koska eivät niitä ymmärrä.

# Vaatimukset ja ohjelmistosuunnittelu

- Periaatteessa, **vaatimusten** pitäisi kuvata mitä järjestelmän pitäisi tehdä ja **(ohjelmisto)suunnittelun (design)** pitäisi kuvata miten järjestelmä tekee sen.
- Käytännössä, ne ovat erottamattomat
  - Järjestelmän arkkitehtuuri saattaa olla suunniteltu **jäsentämään vaatimuksia**;
  - Järjestelmä tulee toimimaan **yhdessä muiden järjestelmien kanssa**, mikä **tuottaa ohjelmistosuunnitteluvaatimuksia**;
  - Tietyn **ohjelmistoarkkitehtuurin** käyttö, jotta ei-toiminnallisia vaatimuksia voidaan toteuttaa, voi olla **sovellusaluevaatimuksena**

# Määritelmät luonnollisella kielellä

- Vaatimukset kirjoitetaan luonnollisen kielen virkkeillä, joihin liitetään kaavioita ja tauluja.
- Tällainen määritelmä on ilmeikäs, intuitiivinen ja universaali, joten vaatimuksia voi ymmärtää käyttäjät ja asiakkaat.

## Esimerkki : Insuliinipumpun ohjelmisto (Luento 1, osa 4)

3.2 Järjestelmä mittaa joka kymmenes minuutti verensokeria ja antaa insuliinia tarvittaessa. *(Verensokeripitoisuus muuttuu kohtuullisen hitaasti, joten tiheämpi mittaaminen ei ole tarpeellista. Harvemmin mittaaminen voi johtaa tarpeettoman korkeisiin verensokeripitoisuuksiin)*

3.6 Järjestelmä testaa oman toimintansa kerran minuutissa. Testattavat asiat, ja mahdollisesti tarvittavat toimenpiteet on kuvattu taulussa 1. *(Järjestelmä voi havaita laitteisto- tai ohjelmisto-ongelmia ja hälyttää käyttäjää, että se mahdollisesti toimii oikein.)*

**Function** Compute insulin dose: safe sugar level.

**Description**

Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

**Inputs** Current sugar reading (r2); the previous two readings (r0 and r1).

**Source** Current sugar reading from sensor. Other readings from memory.

**Outputs** CompDose—the dose in insulin to be delivered.

**Destination** Main control loop.

**Action**

CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.

**Requirements**

Two previous readings so that the rate of change of sugar level can be computed.

**Pre-condition**

The insulin reservoir contains at least the maximum allowed single dose of insulin.

**Post-condition** r0 is replaced by r1 then r1 is replaced by r2.

**Side effects** None.

# Strukturoidut määritelmät

- Requirements elicitation
- [Requirements specification](#)
- Requirements validation
- Requirements change

- Vaatimusmääritelmät kirjoitetaan tarkasti rajatulla standardilla tavalla.
- Toimii joissakin tilanteissa, esimerkiksi sulautettujen valvontajärjestelmien kuvauksissa, mutta voi olla liian kahlitseva kuvaamaan dynaamisessa liiketoimintaympäristössä toimivan järjestelmän vaatimuksia.

# Taulukkomuotoinen määritelmä

- Requirements elicitation
- Requirements specification
- Requirements validation
- Requirements change

- Tukee luonnollisen kielen kuvauksia.
- Erityisen hyödyllinen kun pitää määritellä joukko mahdollisia vaihtoehtoisia toimintavaihtoehtoja.
- Esimerkiksi, insuliinipumppujärjestelmä perustaa laskentansa mitattuihin verensokeritason muutoksiin ja taulukossa määritettyihin rajoihin.

Condition	Action
Sugar level falling ( $r2 < r1$ )	CompDose = 0
Sugar level stable ( $r2 = r1$ )	CompDose = 0
Sugar level increasing and rate of increase decreasing ( $(r2 - r1) < (r1 - r0)$ )	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ( $(r2 - r1) \geq (r1 - r0)$ )	CompDose = round $((r2 - r1)/4)$ If rounded result = 0 then CompDose = MinimumDose

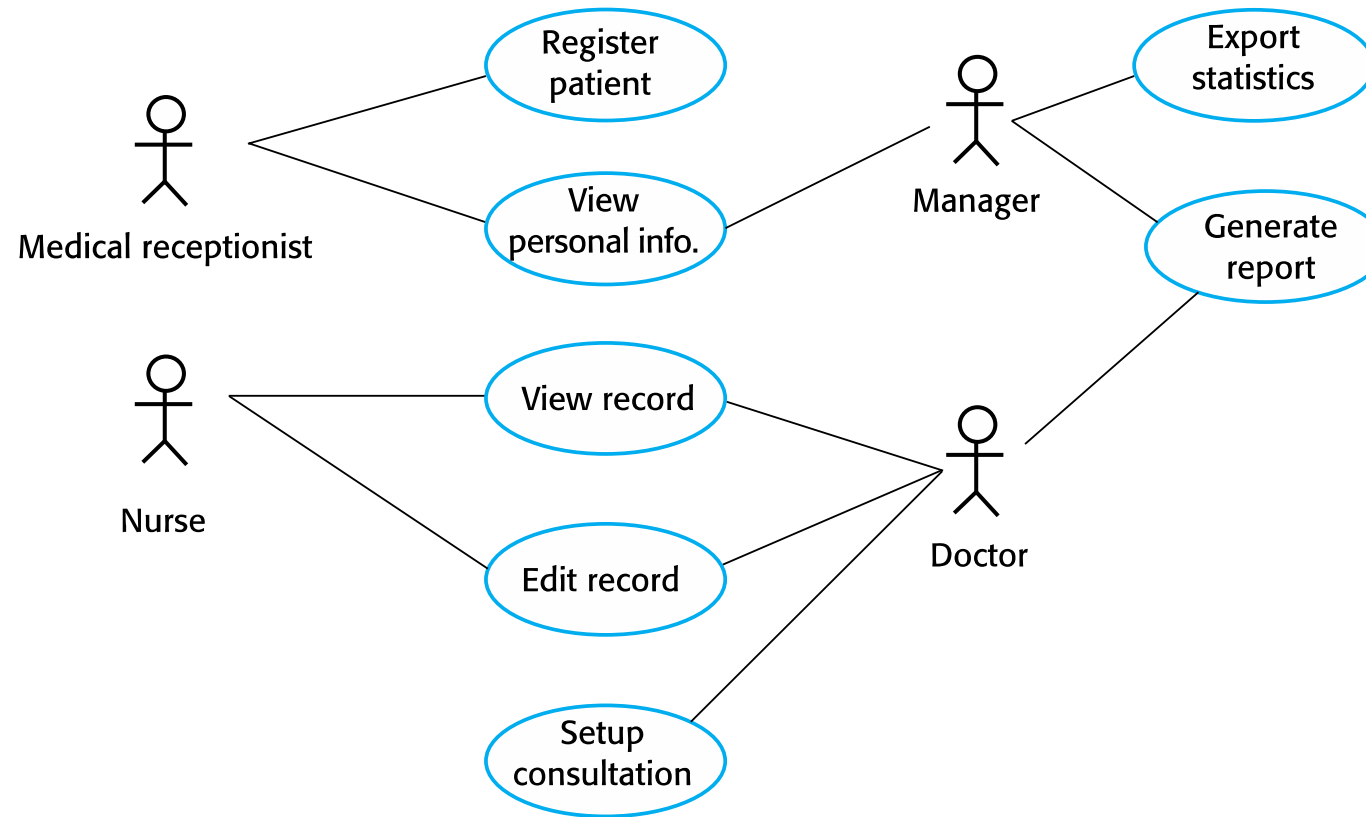
# Käyttötapaukset (Use cases)

- Käyttötapaukset kuuluvat UML-kuvausperheeseen.
  - UML (Unified modelling language) on ohjelmistosuunnittelun eniten käytetty kuvausstandardi.
- Käyttötapaus määrittelee tilanteen, jossa tietojärjestelmän käyttäjä (actor) käyttää tietojärjestelmää.
- Kaikkien käyttötapausten joukko pitäisi kattaa kaikki mahdolliset käyttäjän ja järjestelmän väliset vuorovaikutustilanteet.
- UML:n sekvenssikaavioita (sequence) voi käyttää lisänä käyttötapauksille, jotta voidaan määritellä käyttötapausten toimintasekvenssi.
- Katso luku 5.



- Requirements elicitation
- Requirements specification
- Requirements validation
- Requirements change

# Use cases for the Mentcare system



# Ohjelmistovaatimusdokumentti

- Virallinen kuvaus, mitä järjestelmältä vaaditaan.
- Sisältää sekä käyttäjävaatimusten määritelmät että järjestelmävaatimusten määritelmät.
- Se **ei** ole ohjelmistosuunnitteludokumentti. Niin pitkälle kuin mahdollista, sen tulisi kuvata **mitä** järjestelmän pitäisi tehdä eikä **miten** sen tulisi velvoitteensa tehdä.

# Vaatimusmäärittäydokumentin rakenne

- Requirements elicitation
- Requirements specification
- Requirements validation
- Requirements change

Luku	Kuvaus
Esipuhe	Kuvataan dokumentin oletettu lukijakunta ja versiohistoria mukaan lukien perusteet uusien versioiden luontiin ja tiivistelmän muutoksista, mitä kuhunkin versioon tehtiin.
Johdanto	Kuvataan: <ul style="list-style-type: none"><li>- järjestelmän olemassaolon tarve</li><li>- lyhyesti järjestelmän toiminnot</li><li>- kuinka se toimii muiden järjestelmien kanssa.</li><li>- kuinka järjestelmä sopii käyttävän/rahoittavan organisaation liiketoimintaan ja strategiaan valintoihin.</li></ul>
Sanasto	Määritellään käytetyt tekniset termit. Olisi hyvä, jos ei tehdä oletuksia lukijoiden asiantuntemuksesta.
Käyttäjävaatimusten määrittely	Kuvataan käyttäjälle tarjottavat palvelut. Samoin ei-toiminnalliset järjestelmävaatimukset kuvataan tässä osassa. Käytetään luonnollista kieltä, kaavioita tai muuta notaatiota, jota asiakas ymmärtää. Tuote- ja prosessistandardit, joita pitää noudattaa, selitetään tässä osassa.
Järjestelmäarkkitehtuuri	Kuvataan korkean tason yleisnäkymä halutusta järjestelmäarkkitehtuurista. Uudelleenkäytettävät komponentit tulee olla merkittynä.

Luku	Kuvaus
Järjestelmävaatimusten määrittely	Määritellään yksityiskohtaisemmin toiminnalliset ja ei-toiminnalliset vaatimukset. Myös käyttöliittymät muihin ulkopuolisiin järjestelmiin kuvataan tässä osassa.
Järjestelmämallit	Tässä osassa voi olla graafiset järjestelmämallit kuvaten järjestelmäkomponenttien yhteyttä järjestelmään ja sen ympäristöön. Mallit voivat olla esimerkiksi olio-, tietovuo- tai semanttisia datamalleja.
Järjestelmän evoluutio	Selitetään perustavanlaatuiset oletukset, mihin järjestelmä perustuu ja mihin ennakoitua tulevaisuudessa tehtävät muutokset vaikuttavat siinä. Osa on hyödyllinen järjestelmän kehittäjille, sillä se helpottaa heitä välttämään päätöksiä, jotka todennäköisesti haittaavat järjestelmän muuttamista tulevaisuudessa.
Liitteet	Niiden tulisi olla yksityiskohtaiset kuvaukset esimerkiksi laitteistokonfiguraatioista tai tietokannoista.
Hakemistot	Useita erilaisia hakemistoja voidaan liittää dokumenttiin - kuten luettelo asiasanoista, kaavioluettelo tai toimintoluettelo.