

# Ohjelmistotekniikan menetelmät

Matti Luukkainen

Helsingin Yliopisto, TKTL

Syksy 2016

# Kurssi on johdanto ohjelmistotuotantoon

wikipedian mukaan ohjelmistotuotanto

- ▶ on yhteisnimitys niille työnteon ja työnjohdon menetelmille, joita käytetään, kun tuotetaan tietokoneohjelmia sekä monista tietokoneohjelmista koostuvia tietokoneohjelmistoja.
- ▶ laajasti ymmärrettynä kattaa kaiken tietokoneohjelmistojen valmistukseen liittyvän prosessinhallinnan sekä kaikki erilaiset tietokoneohjelmien valmistamisen menetelmät.
- ▶ kattaa siis kaikki aktiviteetti, jotka tähtää tietokoneohjelmien tai -ohjelmistojen valmistukseen.

## Mallintaminen

- ▶ Ohjelmistotuotantoon liittyy **mallintaminen**, eli kyky tuottaa erilaisia kuvauksia, joita tarvitaan ohjelmiston kehittämisen yhteydessä
- ▶ Mallit toimivat kommunikoinnin välineinä
- ▶ *Mitä ollaan tekemässä, miten ollaan tekemässä, mitä tehtiin?*
- ▶ Kurssi oli aiemmin nimeltään *Ohjelmistojen mallintaminen*, painopiste on hieman muuttunut, mutta mallintaminen on edelleen vahvasti mukana

# Kurssista

- ▶ **Aikataulu ja kurssimateriaali:**

- ▶ <https://github.com/mluukkai/OTM16>

- ▶ **Laskarit:**

- ▶ Aloitetaan jo ensimmäisellä viikolla
  - ▶ Yhteensä 6kpl, 3h per tilaisuus

- ▶ **Arvostelu:**

- ▶ Kurssin kokonaispistemäärä on 36p
  - ▶ Kurssikoe 22p
  - ▶ Laskareista 14p, jotka koostuu...
    - ▶ Paikanpäällä tehtävistä 7p (1p per kerta)
    - ▶ Etukäteen tehtävistä 7p (90% → 7p)
  - ▶ Noin 32p → arvosana 5
  - ▶ Läkipääsy vaatii puolet kurssikokeen pisteistä ja puolet laskaripisteistä, eli ainakin 18p

# Laskarit

## ► Etukäteen tehtävät:

- viikoilla 2-7
- Keskimäärin 6 tehtävää viikossa
- Tehdään etukäteen niin, että vastauksia voidaan tarkastella ryhmissä, eli tulostettuna, läppärillä tai verkossa

## ► Paikanpäällä tehtävät:

- Ryhmätyöskentelyä (myös ensimmäisellä viikolla!)
- Tehdään niin paljon kuin ehtii, mutta työskennellään aktiivisesti
- Laskarit eivät ole paja, paikalla on oltava alusta loppuun!

## ► Laskariajat

- Ryhmä 1: ke 9-12 B221
- Ryhmä 2: ke 14-17 B221
- Ryhmä 3: to 9-12 B221
- Ryhmä 4: to 14-17 B221
- Ryhmä 5: pe 9-12 B221
- Ryhmä 6: pe 12-15 B221

# Ohjelmistotuotantoprosessi

## Miksi prosessi kun voi vain tehdä?

- ▶ Pienissä itselle tehtävissä projekteissa voidaan ohjelmida noudattamatta mitään systematiikkaa
  - ▶ Voidaan helposti häkätä kasaan sovellus, joka *"toimii"*
- ▶ Tämä menetelmä ei toimi isommille, monen hengen projekteissa asiakasta varten tuotetuille ohjelmille
  - ▶ Työn jakaminen tekijätiimiläisten kesken on hankalaa
  - ▶ Jää epäselväksi toimiiko sovellus niin kuin alunperin haluttiin?
  - ▶ Ohjelman rakenteesta tulee epämääräinen ja sen takia laajennettavuus ja ylläpidettävyyys on vaikeaa
- ▶ Ratkaisuna on kehitelty lukuisia erilaisia menetelmiä ohjelmistotuotantoprosessin systematisoimiseksi <sup>1</sup>
- ▶ Mitä menetelmää tulisi käyttää? Hyvä kysymys!

---

<sup>1</sup>[https://en.wikipedia.org/wiki/List\\_of\\_software\\_development\\_philosophies](https://en.wikipedia.org/wiki/List_of_software_development_philosophies)

# Ohjelmistotuotantoprosessin vaiheet

Käytetystä menetelmästä riippumatta ohjelmistotuotantoprosessissa tapahtuu seuraavia aktiviteetteja

## 1. Vaatimusmäärittely

- ▶ Mitä halutaan?

## 2. Suunnittelu

- ▶ Miten tehdään?

## 3. Toteutus

- ▶ Ohjelmoidaan

## 4. Testaus

- ▶ Varmistetaan että toimii niin kuin halutaan

## 5. Ylläpito

- ▶ Korjataan bugeja ja laajennetaan ohjelmistoa

# Ohjelmistotuotantoprosessin vaiheet

## Vaatusmäärittely

Vaatusmäärittelyssä kartoitetaan ja dokumentoidaan **mitä asiakas haluaa**

- ▶ Servitetään sovelluksen **toiminnalliset vaatimukset**
  - ▶ Miten ohjelman tulisi toimia?
- ▶ ja toimintaympäristön asettamat **rajoitteet**
  - ▶ Toteutusympäristö
  - ▶ Suorituskykyvaatimukset
  - ▶ Luotettavuusvaatimukset
  - ▶ Käytettävyys
- ▶ Ei vielä puututa siihen miten järjestelmä tulisi toteuttaa
- ▶ Ei oteta kantaa ohjelman sisäisiin teknisiin ratkaisuihin, ainoastaan siihen miten toiminta näkyy käyttäjälle



# Ohjelmistotuotantoprosessin vaiheet

## Vaatusmäärittely

Esim: Yliopiston kurssinhallintajärjestelmä

- ▶ Toiminnallisia vaatimuksia:
  - ▶ Opetushallinto voi syöttää kurssin tiedot järjestelmään
  - ▶ Opiskelija voi ilmoittautua valitsemalleen kurssille
  - ▶ Opettaja voi syöttää opiskelijan suoritustiedot
  - ▶ Opettaja voi tulostaa kurssin tulokset
- ▶ Toimintaympäristön rajoitteita:
  - ▶ Kurssien tiedot talletetaan jo olemassa olevaan tietokantaan
  - ▶ Järjestelmää käytetään www-selaimella
  - ▶ Toteutus Javalla
  - ▶ Kyettävä käsittelemään vähintään 100 ilmoittautumista minuutissa

# Ohjelmistotuotantoprosessin vaiheet

## Vaatusmäärittely

Toinen esimerkki: Reittiopas

- ▶ Toiminnallisia vaatimuksia:
  - ▶ Käyttäjä voi etsiä kahden osoitteen välisiä liikenneyhteyksiä
  - ▶ Käyttäjä voi etsiä nykyisen sijaintinsa ja jonkun osoitteen välisiä liikenneyhteyksiä
  - ▶ Käyttäjä voi valita minkä tyyppisiä kulkuneuvoja haluaa liikkueessaan käyttää
  - ▶ HSL:n virkailija voi ylläpitää kulkuneuvojen aikataulutietoja
- ▶ Toimintaympäristön rajoitteita:
  - ▶ Järjestelmän voi käyttää web-selaimella
  - ▶ Järjestelmän voi käyttää mobiililaitteelle asennetulla applikaatiolla
  - ▶ Mobiiliapplikaatiot on toteutettu natiivitekniikalla
  - ▶ Järjestelmän käsittelemään vähintään 10000 kyselyä minuutissa

# Ohjelmistotuotantoprosessin vaiheet

## Vaatusmäärittely

- ▶ Jotta toteuttajat ymmärtäisivät mitä pitää tehdä, joudutaan ongelma-aluetta analysoimaan
  - ▶ Esimerkiksi jäsennetään ongelma-alueen käsitteistöä
  - ▶ Tehdään ongelma-alueesta *malli* eli yksinkertaistettu kuvaus
- ▶ Vaatusmäärittelyn päätteeksi yleensä tuotetaan **määrittelydokumentti**
  - ▶ Kirjaa sen mitä ohjelmalta halutaan
  - ▶ Käytetään ohjeena suunnitteluun ja toteutukseen
- ▶ Määrittelydokumentin sijaan määrittely (tai ainakin sen osa) voidaan myös ilmaista ns. hyväksymiskriteereinä tai -testeinä. Tällöin ohjelma toimii "määritelmänsä mukaisesti" jos se läpäisee kaikki määritellyt hyväksymiskriteerit

# Ohjelmistotuotantoprosessin vaiheet

## Ohjelmiston suunnittelu

Miten saadaan toteutettua määrittelydokumentissa vaaditulla tavalla toimiva ohjelma?

Suunnittelussa on useimmiten kaksi vaihetta

### 1. Arkkitehtuurisuunnittelu

- ▶ Määritellään ohjelman rakenne karkealla tasolla
- ▶ Mistä suuremmista rakennekomponenteista ohjelma koostuu?
- ▶ Miten komponentit yhdistetään, eli komponenttien väliset rajapinnat

### 2. Oliosuunnittelu

- ▶ yksittäisten komponenttien suunnittelu

Suunnittelun lopputuloksena on yleensä suunnitteludokumentti

## Ohjelmiston suunnittelu

- ▶ **Suunnitteludokumentti**
  - ▶ Ohje toteuttajille
  - ▶ Joskus/usein suunnittelu- ja ohjelmointivaihe ovat niin kiinteästi sidottuna toisiinsa, että tarkkaa suunnitteludokumenttia ei tehdä
  - ▶ Joskus koodi toimii dokumenttina
- ▶ **Mallit liittyvät vahvasti suunnitteluun!**
  - ▶ Arkkitehtuurikuvaus
    - ▶ Järjestelmän alikomponentit
    - ▶ Komponenttien väliset rajapinnat

