Kuvaus- ja mallintamismenetelmät

Lähtökohdat, tavoitteet ja tekniikat

Opintojakson oppimistavoitteet

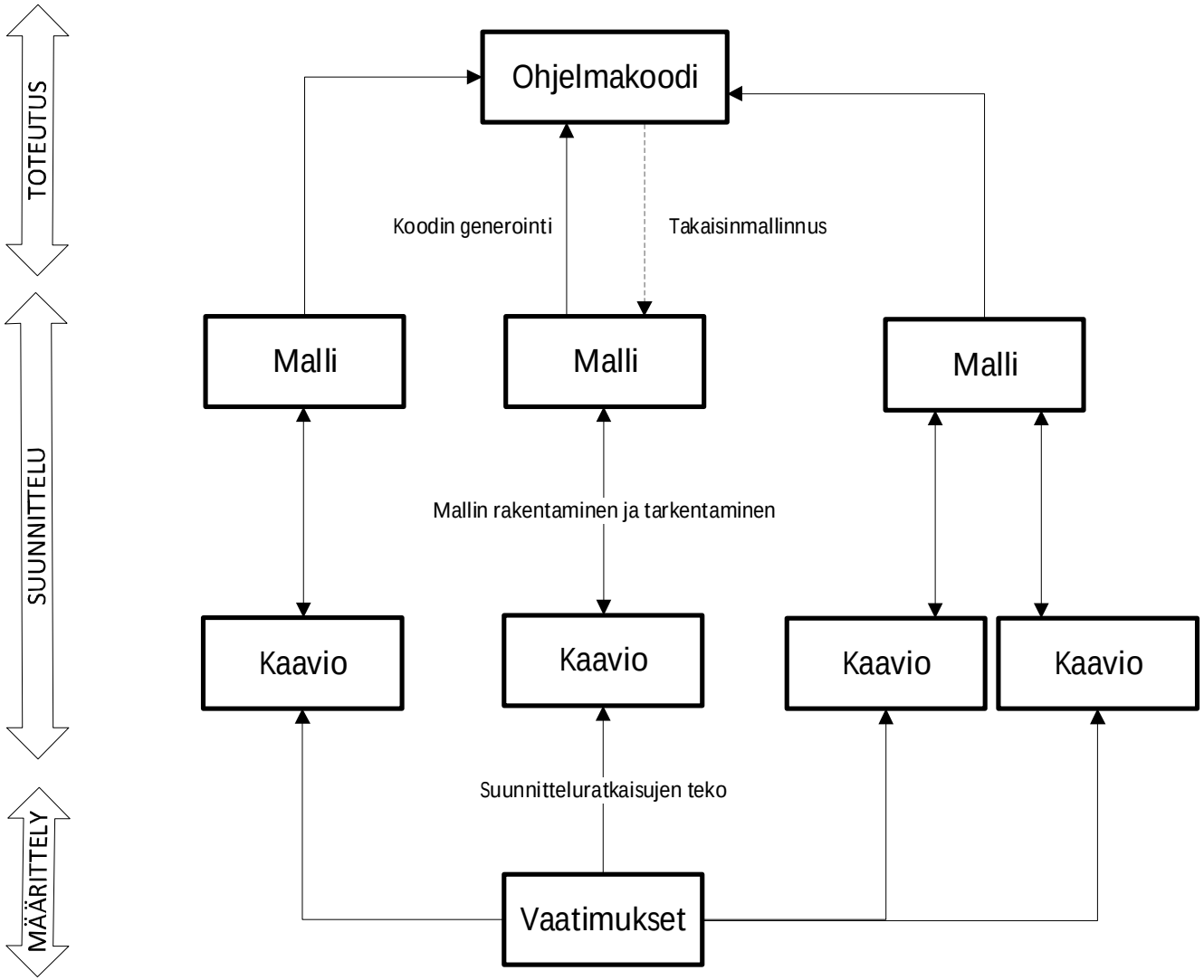
- Opintojakson suoritettuasi:
  1. ymmärrät datan ja ohjelmistojen mallinnuksen tavoitteet ja tarpeen
  2. tunnet ER- ja UML-kaaviotekniikat ja niiden käyttökohteet
  3. osaat laatia datan rakennetta sekä ohjelmistosi rakennetta ja toimintaa kuvaavia kuvauksia ja malleja
  4. osaat generoida mallinnetusta datasta relaatiotietokannan.
  5. osaat tulkita toisten kehittäjien tuottamia kuvauksia ja malleja
- Opintojaksolla opit siis
  - suunnittelemaan ja kuvamaan ohjelmistojen ratkaisuja
  - esittelemään ja kommunikoimaan ratkaisuja ammattimaisesti ja myös toisten kehittäjien ymmärtämällä tavalla
  - käyttämään ER- ja UML-kaaviotekniikoita omassa ohjelmistoprojektiasi sitä hyödyttävällä tavalla

Mitä on mallintaminen?

- Mallintaminen on datan rakennetta tai ohjelmiston rakennetta ja toimintaa kuvaavien esitysten luomista.
- Malli on abstrakti esitys datasta tai ohjelmistotuotteesta valituista näkökulmista.
  - Esimerkki mallintamisesta on ohjelmiston luokkarakenteen suunnittelu.
  - Luokkarakenne on vain yksi mahdollinen näkökulma ohjelmistoon.
- Mallintaminen on keskeinen ja erottamaton osa ohjelmistotuotantoa.

Mitä on kuvaaminen?

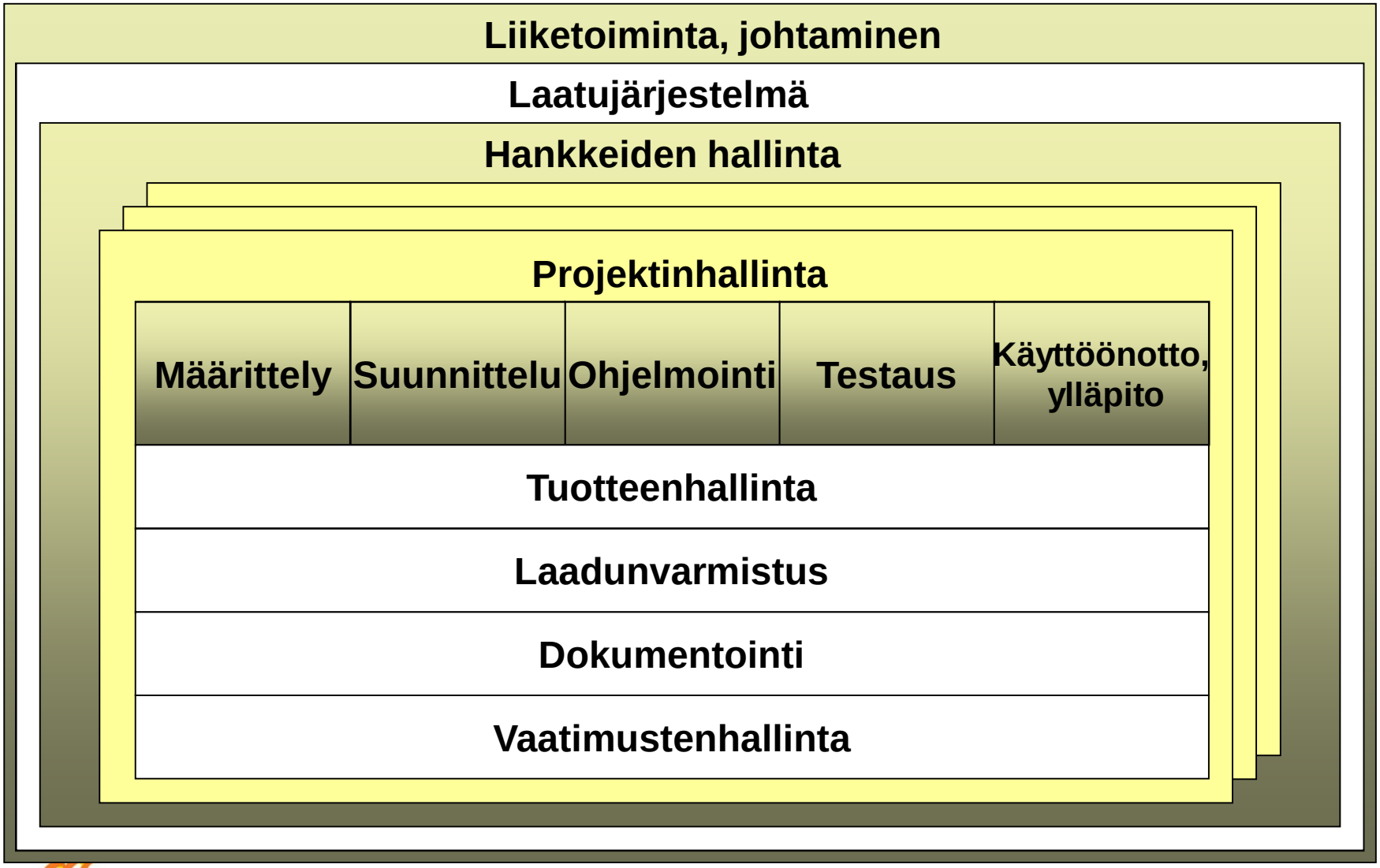
- Kuvaamisella tarkoitetaan mallin esittämistä visuaalisessa muodossa.
  - Esimerkiksi mallissa suunniteltu luokkarakenne voidaan esittää kuvana luokista, ns. luokkakaaviona.
- Kuvaustekniikoita tarvitaan
  1. Mallien luomiseksi ja muokkaamiseksi
  2. Malleista keskustelemiseksi ja niistä viestimiseksi


TOTEUTUS
SUUNNITTELU
MÄÄRITTELY
Ohjelmakoodi
Koodin generointi
Takaisinmallinnus
Malli
Malli
Malli
Mallin rakentaminen ja tarkentaminen
Kaavio
Kaavio
Kaavio
Kaavio
Suunnitteluratkaisujen teko
Vaatimukset

Ohjelmisto ja tietojärjestelmä

- Ohjelmistolla tarkoitetaan kokonaisuutta, joka koostuu
  - Tietokoneessa ajettavista toisiinsa liittyvistä ohjelmista
  - Niiden tarvitsemista tiedoista
  - Niitä tukevasta dokumentaatiosta
- Tietojärjestelmällä tarkoitetaan liiketoimintatavoitteiden saavuttamiseksi luotua kokonaisuutta tiedon
  - Luomiseksi ja keräämiseksi
  - Tallentamiseksi ja varastoimiseksi
  - Käsittelemiseksi ja siirtämiseksi
- Tietojärjestelmä koostuu
  - Toisiinsa liittyvistä ohjelmistoista
  - Datan säilytys- ja tallennusratkaisuista (tietokannat)
  - (Ihmisistä)

Ohjelmistotuotanto

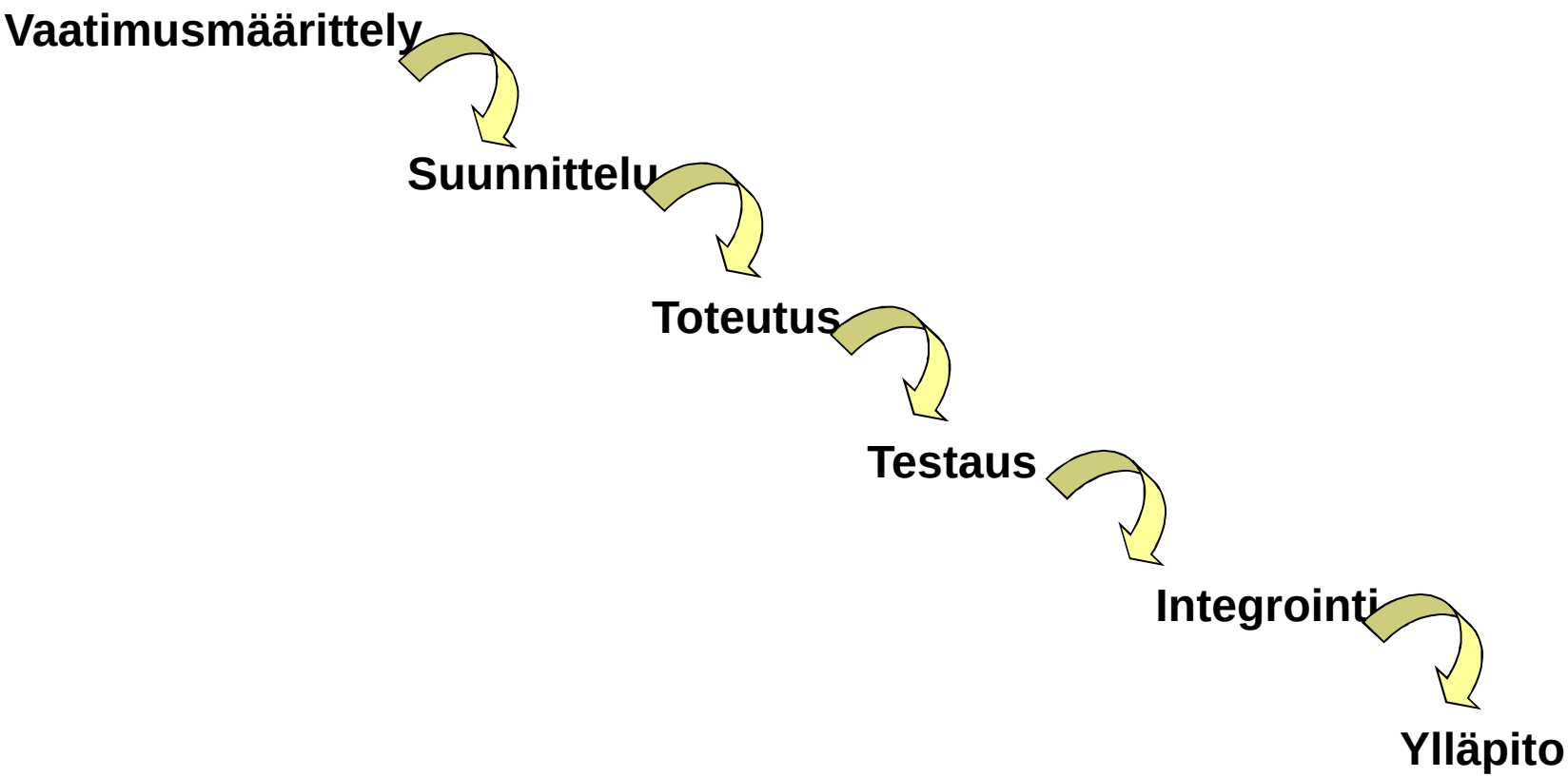


Ohjelmiston elinkaari ja vaihejakomallit

- Elinkaari
  - Aika joka kuluu ohjelmiston kehitystyön aloituksesta sen käytöstä poistamiseen
- Vaihejakomalli
  - Tapa, jolla ohjelmiston elinkaari (kokonaisuudessaan tai kehitystyön osalta) jaetaan toisiaan seuraaviin vaiheisiin.
  - Perinteinen vesiputousmalli vs. iteratiiviset mallit

Huomaa, että malli-termiä käytetään moneen tarkoitukseen. Tässä sillä ei tässä tarkoiteta ohjelmistotuotteesta laadittavaa mallia, vaan tapaa organisoida ohjelmistoprojekti.

Vesiputousmalli

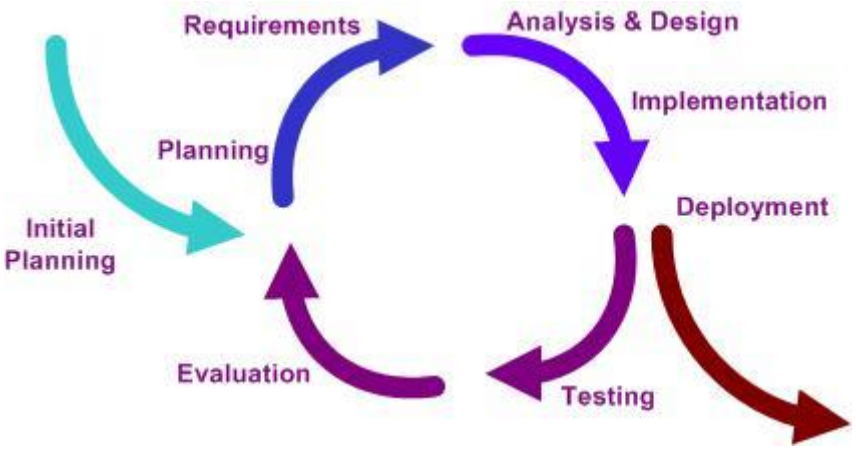


Muita vaihejakomalleja

- Prototyyppimalli
- Evolutiiviset mallit
- Spiraalimalli
- Ketterät menetelmät

Ketterät (*agile*) menetelmät

- Ketterät menetelmät muuttavat näkökulman ohjelmistotuotantoon vesiputousmalliin verrattuna.
  - Näkökulma asiakkaassa eikä projektinhallinnassa
  - Iteratiivista
    - tuote paranee kierros kierrokselta
  - Inkrementaalista
    - kokonaisuus koostetaan osista
  - Minimoidaan laatuvelan ja teknisen velan kasautumista.



Kuvan lähde: Wikimedia Commons

Ketterä manifesti

- http://agilemanifesto.org/
- Ketterä manifesti julkituo ketterän ohjelmistokehityksen arvot:
  1. yksilöt ja vuorovaikutus vs. prosessit ja työkalut
  2. toimiva ohjelmisto vs. kattava dokumentaatio
  3. vuorovaikutus asiakkaan kanssa vs. sopimusneuvottelut
  4. muutoksiin reagoiminen vs. suunnitelman seuraaminen

- Käytännössä ketterissä menetelmissäkin on sisäänrakennettuna piirteitä vesiputousmallista.
  – Yleensä ainakin jonkinlainen määrittelyvaihe.

Ketterä toteutusmalli 1: Scrum

- Projektiorganisatorinen näkökulma: minimoidaan työ, joka ei edistä projektin etenemistä.
- Roolitus
  - *Scrum team:* kehitystiimin jäsenet
  - *Scrum master:* vastaa kehitystiimin jäsenten ongelmien ratkomisesta.
  - *Product owner:* tuoteomistaja, joka päättää, mitä kehitystiimi tekee ja mitä ei tee (tuotteeseen). Ei yleensä koodaa.

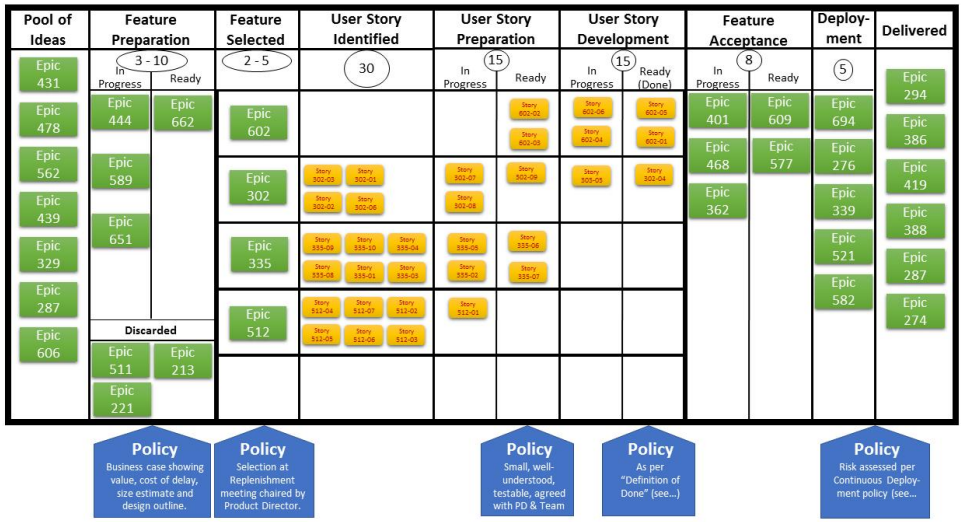
Ketterä toteutusmalli 2: Lean

- Näkökulma lisäarvon tuottamisessa asiakkaalle: minimoidaan työ, joka ei tuota asiakkaalle mitattavaa arvoa.
- Toyotan kehittämä.
- Minimoimaan ”*jäte*” (*waste*) eli kaikki, mikä ei tuota lisäarvoa asiakkaalle:
  - Keskeneräinen työ
  - Ylimääräiset ominaisuudet
  - Ylimääräiset prosessit (esim. dokumentit)
  - Kontekstin vaihdot
  - Odotus ja viivästykset
  - Työn siirto toiselle henkilölle
  - Viat

Lähde:
https://agilecoachdiaries.wordpress.com/2017/01/19/lean-software-development-7-wastes-of-software-development/

Ketterä toteutusmalli 3: Kanban

- Lähtökohtina kehitystyön visualisointi ja jatkuva toimitus.
- Visualisoinnin tavoitteena on, että kehittäjillä säilyy käsitys kokonaisuudesta ja osien roolista siinä.
- Jatkuvassa toimituksessa (CD; *continuous delivery*) kehitettävä tuote pidetään koko ajan jakelukelpoisena.
  - Tämä edellyttää automatisoitua koontia ja testausta.
  - Uusi, tuotettu koodi voi päätyä uuteen jakeluversioon periaatteessa milloin vain.



Poistuvia ketteriä toteutusmalleja

- RUP
  - *Rational Unified Process*
  - Nykyään IBM:n omistaman Rationalin luoma kaupallinen järjestelmänkehitysprosessi
  - Tällä kurssilla opittava UML kehittyi huomattavasti RUP:n myötä.
- XP
  - *Extreme Programming*
  - kokoelma ns. hyviksi havaittuja menetelmiä (parikoodaus, jatkuva testaus jne.)
  - Ideat elävät muissa toteutusmalleissa.

Ohjelmistoprojektien haasteita

- Tiimityön yleiset haasteet
  - Kieli, kommunikaatio, näkemykset, osaaminen
- Korkea epäonnistumisprosentti
- Budjetti, aikataulu ja laatu on vaikeaa saada samanaikaisesti pitämään.
  - PMI:n raportin[1] (2017) mukaan
    - 14% epäonnistuu täysin
    - 49% ei valmistu ajallaan
    - 43% ylittää budjettinsa
    - 31% ei saavuta tarkoitustaan
- Kuvaus- ja mallintamismenetelmien hallinta on ohjelmistoprojektin toimivuuden edellytyksiä.

1) Lähde: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf

Orientaatioharjoitus: Case Lomahotelli

- Orientaatioharjoituksen tavoitteena on kartoittaa kuvaus- ja mallintamistarpeiden moninaisuutta.

1. Tutustu opettajan antamaan tietojärjestelmän tekstikuvaukseen.
2. Keksi esimerkkejä kysymyksistä, joita sinulle herää järjestelmään perehtyvänä IT-asiantuntijana.
   - Kirjaa kysymyksesi opettajan esittämään JamBoard-pohjaan.
   - Pohjassa on valmiina muutamia esimerkkejä. Keksi lisää erilaisia kysymyksiä!
3. Lopuksi ryhmittelemme kysymyksiä ja pohdimme yhdessä, minkälaisin tekniikoin kysymyksiin voidaan esittää selkeä vastaus.