

# Johdatus(ta) ohjelmistotekniikkaan 2006

Tommi Kärkkäinen  
Jyväskylän yliopisto  
tietotekniikan laitos

## Agendaa

1. IT:n kehitystrendeistä
2. Ohjelmista
3. Ohjelmien kehittämisestä
4. Ohjelmistotekniikasta

# 1. Trendeistä (I)

- ICTs (Information and Communication Technology) play an increasingly important role in the economy.
- ICTs account for a large share of investment and contribute significantly to output and productivity growth.
- Despite current cyclical difficulties, trend growth of the ICT sector remains strong.  
(huom: johtopäätökset hyvästä koulutusalaasta!)
- The ICT sector is highly globalised in an increasingly globalised world economy.
- The focus of international investment is shifting from manufacturing towards services.
- Software is one of the most rapidly growing and evolving sectors in OECD countries.
- Strong growth in the software sector is due to its increasingly crucial role in the ICT sector and the economy.
- Software that underpins network integration, interconnection and compatibility will be essential... and software-related patenting is increasing rapidly.
- Trade in software is dynamic but difficult to measure.
- E-commerce is growing, but is still in its infancy, especially among consumers.  
(huom: lyhyen aikavälin muutokset yliarvioidaan ja pitkän aikavälin aliarvioidaan!)
- The demand for ICT skills continues to grow, creating concerns about possible labour shortages and gaps in worker skills.
- Governments, firms and educational institutions in OECD countries are taking measures to meet changing skill demands in the IT workforce.
- New computing potential and communication channels are being developed and are proliferating.

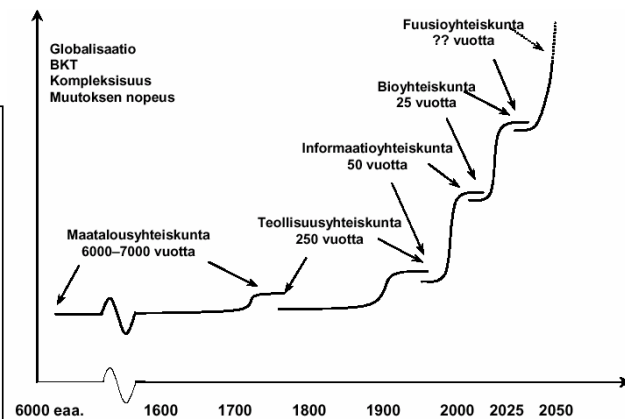
# Trendeistä (II)

| Informaatio- ja<br>kommunikaatioteknologia                              | Bioteknologia                                      | Materiaali- ja<br>nanoteknologia                                   |
|-------------------------------------------------------------------------|----------------------------------------------------|--------------------------------------------------------------------|
| Edistyneen tiedon varastointi<br>(advanced data storage)                | Keinoelimet<br>(artificial organs)                 | Bioyhensopivat polymeeripinnat<br>(biocompatible polymer surfaces) |
| Keinoäly (artificial intelligence)                                      | Biosirut (biochips)                                | Polttoainekennnot (fuel cells)                                     |
| Tietokoneistettu terveydenhuolto<br>(computerized healthcare)           | Biomimetikka<br>(biomimetics)                      | Funktionaaliset polymeerit<br>(functional polymers)                |
| Etäopetus ja etäoppiminen<br>(distance learning)                        | Kloonaus (cloning)                                 | Älymateriaalit<br>(intelligent materials)                          |
| Elektroninen paperi<br>(electronic paper)                               | Geneettinen muuntelu<br>(genetic engineering)      | Miniaturisaatio<br>(miniaturization)                               |
| Modulaariohjelmistot<br>(modular software)                              | Geeniterapia<br>(genetic therapy)                  | Sensorit, havaitsimet<br>(sensors)                                 |
| Neuraaliverkot<br>(neural networks)                                     | Kohdennetut lääkkeet<br>(targeted pharmaceuticals) | Suprajohtavat materiaalit<br>(superconducting materials)           |
| Optiset tietokoneet<br>(optical computers)                              |                                                    |                                                                    |
| Älyagentit (intelligent agents)                                         |                                                    |                                                                    |
| Kaikkialla läsnäoleva tietotekniikka<br>(ubiquitous computing)          |                                                    |                                                                    |
| Virtuaalitodellisuuden sovellukset<br>(applications of virtual reality) |                                                    |                                                                    |

## Trendeistä (III)

### Tulevaisuusammatteja:

1. Keinoelinten suunnittelija
2. Tekoälyn konsultti
3. Bioelektroniikan suunnittelija
4. Bioinformaatikko
5. Hypekonsultti
6. Kyberluokittelija
7. Geeniterapiakonsultti
8. Geoinformaatikko
9. Nanoteknologiakonsultti
10. Yksinkertaistusasiantuntija
11. Älytalojen suunnittelija
12. Sosiaalisten verkostojen analyytikko
13. Virtuaalilääkäri
14. Visualisoinnin asiantuntija
15. Web-puutarhuri



Kuva 2. Yhteiskunnallisen muutoksen suuret aallot (Mannermaa 2002).

- IT:stä tulee keskeinen osa perusinfrastruktuuria jokaisella elämänalueella
- IT-ammattilaisia tarvitaan enenevässä määrin jokaisella elämänalueella
- (huom: lista muodostettu nykyisten trendien pohjalta tämän hetken asiantuntijoiden toimesta eli kriittisyyttä mm. täsmällisiin ammattinimikkeisiin!)

J OPM, Avainteknologiat ja tulevaisuus - Yhteiskunnallisia tarkasteluja nousevien teknologioiden ja kvalifikaatioiden yhteyksistä, 2003

## Trendeistä (IV): yhteenvetoa

- IT/OT:llä keskeinen rooli yhteiskuntakehityksessä meillä ja muualla
- uudet alat, ammatit, työtehtävät ym. perustuvat uusiin ohjelmistoihin, jotka ovat osa uusia, entistä laajempia tieto- ja informaatiojärjestelmiä
- Internetin rooli välitys-, jakelu- ja markkinointikanavana kasvaa; hajautettujen, tietoturvallisten ja kaikkialla läsnä olevien sovellusten määrä lisääntyy
- tietoa täytyy hallita ja muokata älykkäästi sekä tarjota sitä jalostuneessa muodossa, jalostuneina palveluina käyttäjien jokapäiväisten rutiinien helpottamiseksi ja päätöksenteon tueksi
  - tarve ja kysyntä monimutkaisille ohjelmistoille kasvaa taatusti!
- Huom: Ohjelmistoihin ja niiden kehitykseen liittyviä työ- ja liiketoimintamahdollisuuksia yksittäisestä kehittäjästä, konsultista ja kouluttajasta esim. yli 100.000:n työntekijän yrityksen teknologiajohtajaksi yllin kyllin!
  - Avoin ja vapaa ohjelmistokehitys! (mm. muuttaa (liike)toiminta- ja oppimismalleja ja -mahdollisuuksia...)

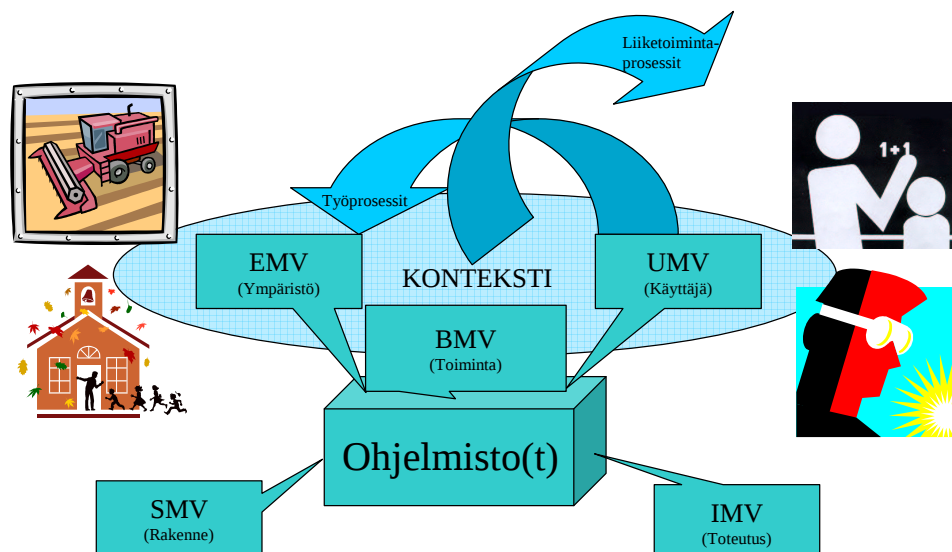
## 2. Ohjelmista

- Mainitse jokin sovellus/ohjelma. Onko se sinusta hyvä vai huono? Miksi?

- Google

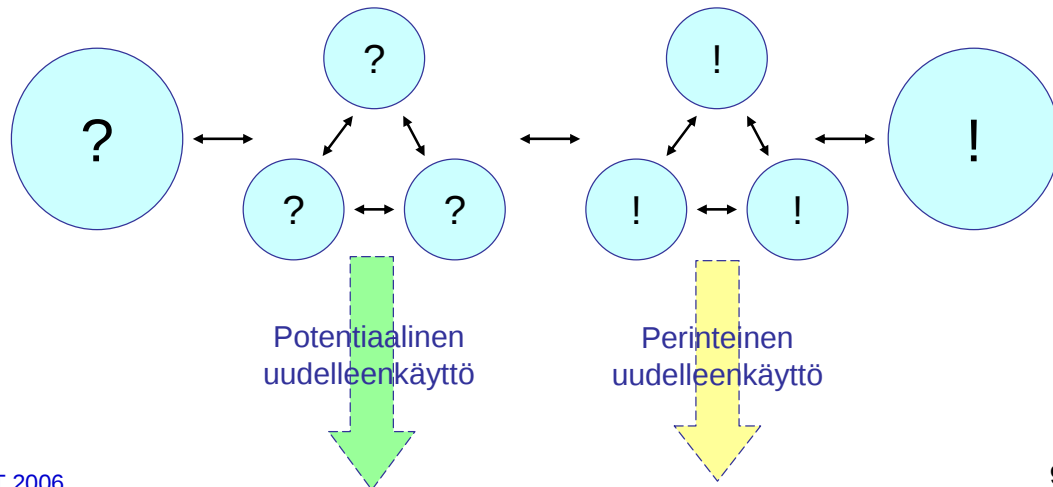
- Skype

## 3. Ohjelmien kehittämisestä: oikeat ohjelmistot oikeille käyttäjille oikeisiin ympäristöihin!

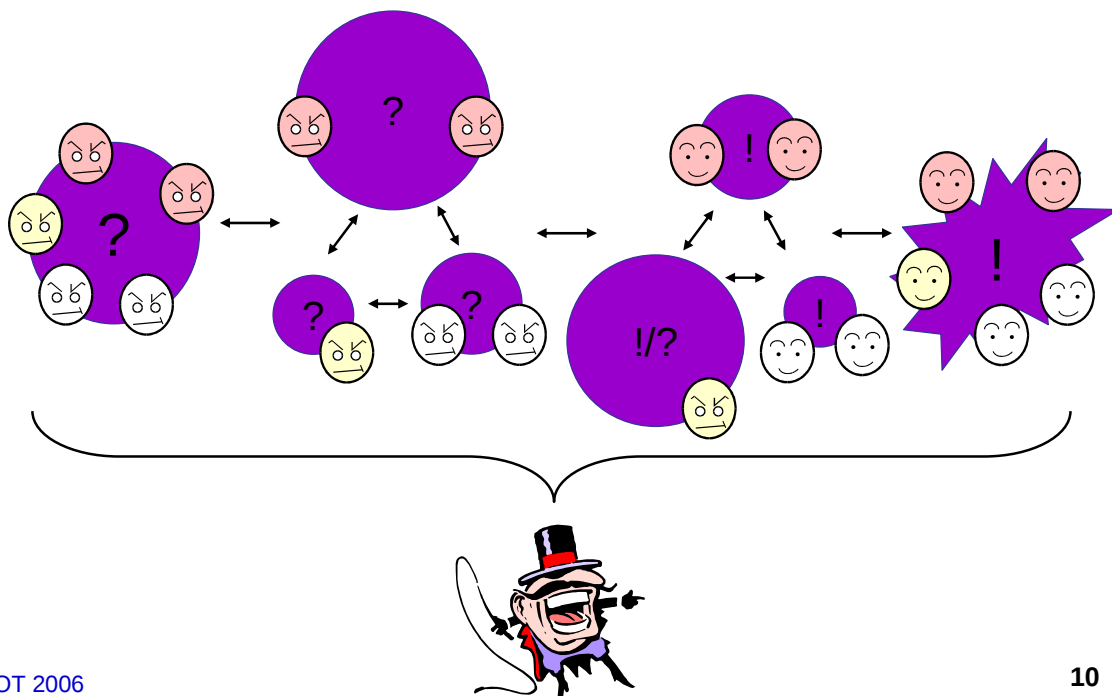


## OK (II): Ongelmanratkaisuanalogia

1. Mikä ongelma halutaan ratkaista? (perus(käyttö)tarkoitus)
2. Miten ratkaistava ongelma jakaantuu osaongelmiin? (vaatimukset)
3. Miten osaongelmat ratkaistaan? (suunnittelu ja toteutus)
4. Mikä saadaan alkuperäisen ongelman ratkaisuksi? (osien kokoaminen)



## OK (III): Ohjelmistoprojekti



## OK (IV): Kaaoshan siitä saattaa syntyä

| Project Success Factors            | % of Responses |
|------------------------------------|----------------|
| 1. User Involvement                | 15.9%          |
| 2. Executive Management Support    | 13.9%          |
| 3. Clear Statement of Requirements | 13.0%          |
| 4. Proper Planning                 | 9.6%           |
| 5. Realistic Expectations          | 8.2%           |
| 6. Smaller Project Milestones      | 7.7%           |
| 7. Competent Staff                 | 7.2%           |
| 8. Ownership                       | 5.3%           |
| 9. Clear Vision & Objectives       | 2.9%           |
| 10. Hard-Working, Focused Staff    | 2.4%           |
| Other                              | 13.9%          |

1995

| Project Challenged Factors                  | % of Responses |
|---------------------------------------------|----------------|
| 1. Lack of User Input                       | 12.8%          |
| 2. Incomplete Requirements & Specifications | 12.3%          |
| 3. Changing Requirements & Specifications   | 11.8%          |
| 4. Lack of Executive Support                | 7.5%           |
| 5. Technology Incompetence                  | 7.0%           |
| 6. Lack of Resources                        | 6.4%           |
| 7. Unrealistic Expectations                 | 5.9%           |
| 8. Unclear Objectives                       | 5.3%           |
| 9. Unrealistic Time Frames                  | 4.3%           |
| 10. New Technology                          | 3.7%           |
| Other                                       | 23.0%          |

### CHAOS Ten

|                             |           |
|-----------------------------|-----------|
| User Involvement            | 20 Points |
| Executive Support           | 15 Points |
| Clear Business Objectives   | 15 Points |
| Experienced Project Manager | 15 Points |
| Small Milestones            | 10 Points |
| Firm Basic Requirements     | 5 Points  |
| Competent Staff             | 5 Points  |
| Proper Planning             | 5 Points  |
| Ownership                   | 5 Points  |
| Other                       | 5 Points  |

1999

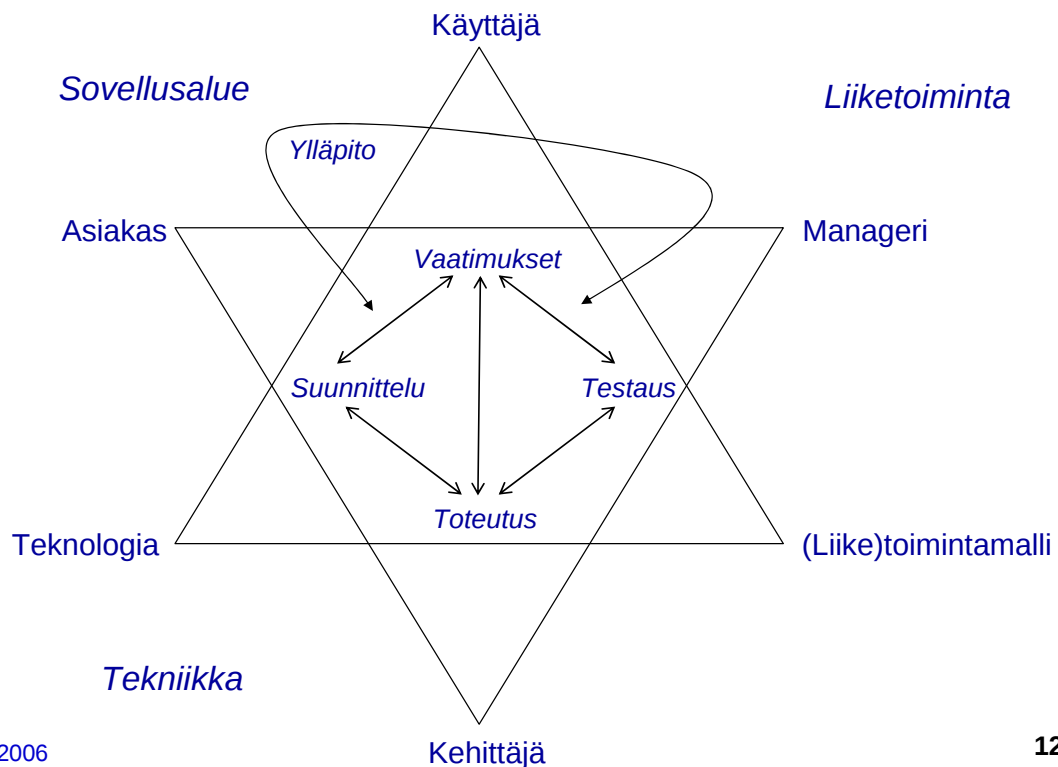
Glimps of 2004: [q3-spotlight.pdf](#)

JOT 2006

IT executive manager surveys (so-called Chaos reports), The Standish Group

11

## OK (V): Rooleja ja näkökulmia

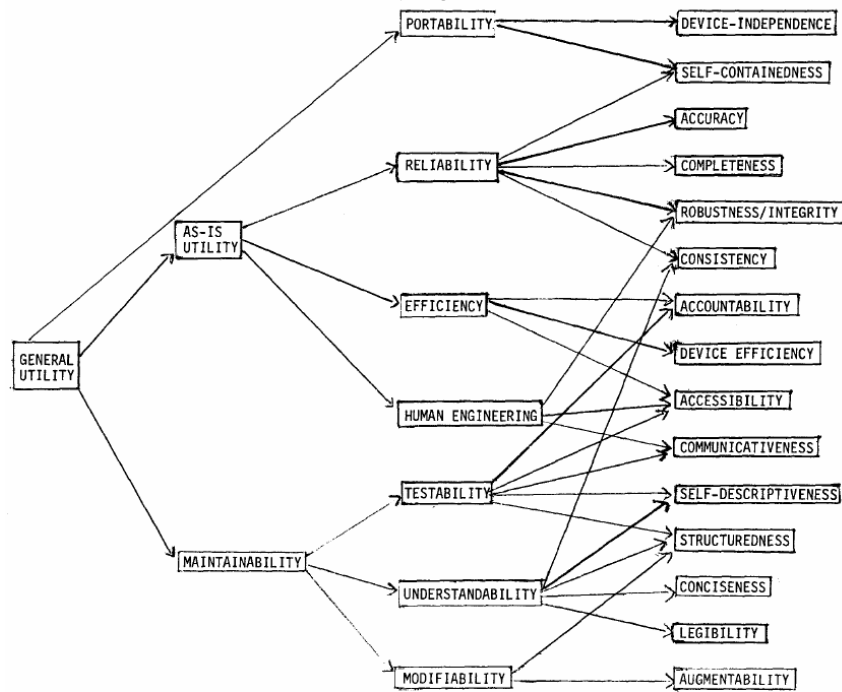


JOT 2006

12

## OK (VI): Laadun kriteerejä (ja näkökulmia)!

Fig. 1 - Software Quality Characteristics Tree



JOT 2006

Boehm, Brown, and Lipow: "Quantitative Evaluation of Software Quality", IEEE Software, 1977

13

## 4. Ohjelmistotekniikasta

- Termi (software engineering, SE) (ja ala) juontaa juurensa 1968 järjestettyyn NATO-konferenssiin, jossa ohjelmistotuotannon todettiin olevan kriisissä – laajojen ohjelmistojen tuottaminen oli vaikeaa, ne olivat pullollaan virheitä, niiden toimitusajat venyivät ja valmistuskustannukset karkasivat käsistä. (vrt. Kaaos-raportteihin)
- Kuten SE:n historiaa käsitelleen seminaarin <http://www.dagstuhl.de/9635/> johdannossa sanotaan:

*In August 1996 about a dozen historians met with about a dozen computer scientists to discuss the history of software engineering. The term software engineering has been deliberately chosen as being provocative at the 1968 NATO Conference on Software Engineering. This notion was meant to imply that software manufacture should be based on the types of theoretical foundations and practical disciplines that are established in the traditional branches of engineering. This need was motivated by the so-called software crisis. Ever since, the debate whether such a crisis exists has continued within the software engineering community. It is a crucial question, because if the answer is yes, software engineering may not be called an engineering discipline yet. If the answer were no, the question would be, what is it that constitutes this discipline.*

- Ammatinimike Software Engineer ja Software Engineering tieteen- ja tutkimusalana ovat eri asioita
- Termi ISD (Information System Development, Tietojärjestelmien kehittämismenetelmät) sotkeutuu ohjelmistokehitykseen&-prosesseihin historiallisista syistä, vaikka TJ-elinkaari prosessi (ja siten toiminnan fokus) on nykykäsityksen mukaan ylemmällä abstraktiotasolla:
  - Planning, Development (SE process; Requirements as glue), Deployment, and Evolution
  - Esim. ylläpidosta puhuttaessa eri abstraktiotasot menevät usein sekaisin (Haikala, Märijärvi)
- Ohjelmistokehityksellä ei ole vakaata teoriapohjaa (toisin kuin OS, PL, DS&AL ym. teknisissä jutuissa), vaan käsitykset "laadukkaasta toiminnasta" perustuvat käytännön kokemuksiin! (konteksti?)

JOT 2006

14

## Ohjelmistotekniikasta (II)

- SE:tä on (tieteenalana) määritelty vuosien kuluessa monella tavalla:

“The establishment and use of sound engineering principles (methods) in order to obtain economically software that is reliable and works on real machines“

(Bauer, 1972)

“Software Engineering is that form of engineering that applies the principles of computer science and mathematics to achieving cost-effective solutions to software problems“

(CMU/SEI-90-TR-003, 1990)

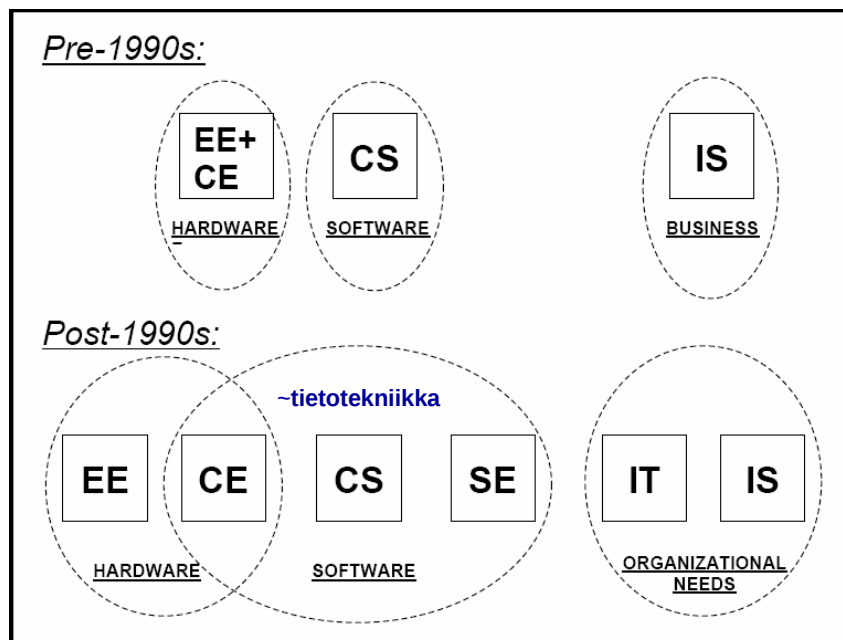
“Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software“

(IEEE Standard Glossary for Software Engineering Terminology, 1990)

“SE applies computer science to practical problems“

(Robert L. Glass, IEEE Software, 2003)

## Ohjelmistotekniikasta (III)



Electrical Engineering  
Computer Engineering  
Computer Science  
Software Engineering  
Information Technology  
Information Systems

## Ohjelmistotekniikasta (IV): yhteenvetoa

- monien kehitysvaiheiden tuloksena, "jättäiläisten harteilla";  
kts. esim. <http://www.sdm.de/en/it-wissen/themen/sdmkonf-2001/index.html>
- Brooks (1987) "No silver bullet: Essence and accidents of software engineering"  
➤ on vaikeaa (ellei mahdotonta) luoda "kaiken ratkaisevia" ohjelmistokehitystekniikoita
- sillan- ja talonrakennus(insinööri)tieteisiin nähden edelleen suuria eroja, erityisesti kvantisoinnissa
  - mutta meillähän kaikki on tietokoneella alusta loppuun?
- (näyttävien) ohjelmien tekeminen nykyvälineillä liian helppoa – liikenteessä vaikka minkälaisia "ohjelmistoja"
- ohjelmistokehitys- ja suoritusympäristön kokonaisvaltainen ymmärtäminen (ihmiset, softat ja raudat) ja jäsentäminen vaativa homma
- Ydinosaatko ohjelmoida ja/vai ohjelmointikieltä ja/vai uusinta sovelluskehitystä ja/vai palveluorientoituneita arkkitehtuuria ja/vai laadunhallintajärjestelmiä ja/vai projektinhallintalakanan käyttöä ja/vai...
- Joo, kaikkea pitää osata (tai oikeastaan riittää osata oppia ja tajuta), mutta silti yhtä tärkeää onnistuneen ohjelmistokehityksen varmistamiseksi on tietää **kuka tekee, mitä tekee, miksi tekee, miten tekee ja milloin tekee**  
⇒ **kehityksen organisointi ja prosessimallit väline- ei itseisarvona!**

## Eka viikon luentotehtävä:

- Etsi IEEE:n standardit (yliopiston verkosta käsin), sieltä "IEEE Standard Glossary of Software Engineering Terminology 1990" ja kuvaa sen avulla seuraavat termit suomeksi omin sanoin:
  - application software
  - incremental development
  - maintainability
  - modularity
  - performance requirement
  - portability
  - test case
  - traceability
  - validation
  - verification
  - verification and validation
- Unified Process –ohjelmistoprosessiin liittyen etsi kuvaukset seuraaville käsitteille:
  - use case driven
  - architecture-centric
  - iterative and incremental
- Vastausformaatti: tekstitiedosto, jonka muotoilu  
[engl.termi]  
[suom.termi]  
<vapaaamuotoinen selitys suomeksi>
- Palautus: Tiedoston URI Korppiin