

Johdatus(ta) ohjelmistotekniikkaan 2005

Tommi Kärkkäinen
Jyväskylän yliopisto
tietotekniikan laitos

Agenda

1. IT:n kehitystrendeistä
2. ohjelmistotekniikasta ja –prosesseista
3. Näkökulmia ohjelmistokehitykseen

1. Trendeistä...

- ICTs (Information and Communication Technologies) 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.

JOT 2005 (OECD *Information Technology Outlook – ICTs and the Information Economy*, 2002) **3**

... trendeistä...

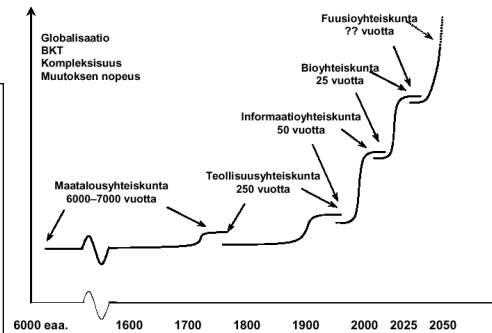
Informaatio- ja kommunikaatioteknologia	Bioteknologia	Materiaali- ja nanoteknologia
Edistyneellinen tiedon varastointi (<i>advanced data storage</i>)	Keinoelimet (<i>artificial organs</i>)	Bioyhteensopivat polymeeripinnat (<i>biocompatible polymer surfaces</i>)
Keinoäly (<i>artificial intelligence</i>)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

(OPM, *Avainteknologiat ja tulevaisuus - Yhteiskunnallisia tarkasteluja nousevien teknologioiden ja kv alifikaatioiden yhteyksistä*, 2003)

... trendeistä...

Tulevaisuusammattaje:

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

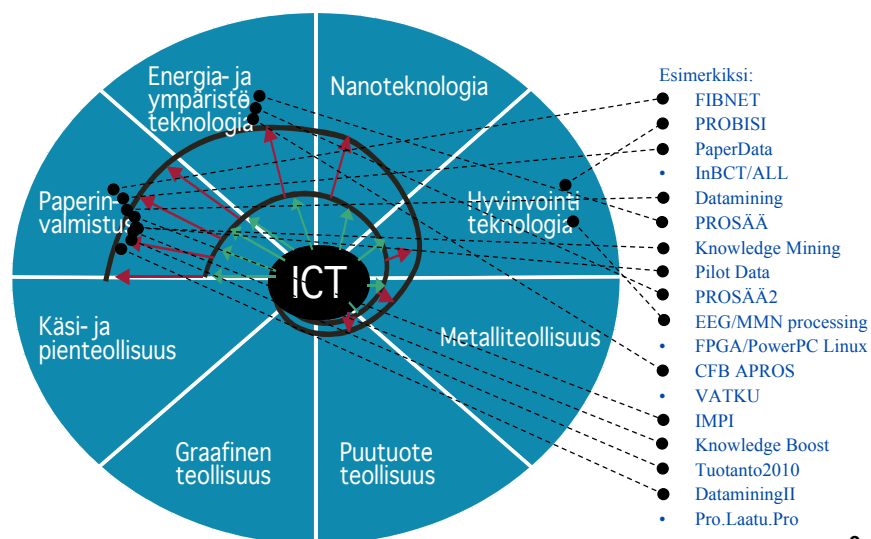


Kuva 2. Yhteiskunnallisen muutoksen suuret aallot (Mannermaa 2002).

- 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 täsmällisiin ammattinimikkeisiin!)

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

IT/OT:n horisontaalisuus: Tieto- ja viestintäteknologian soveltaminen Keski-Suomen teollisuudessa - alueellinen teknologiastrategia



Välitilin päätös:

- IT/OT:llä keskeinen rooli yhteiskuntakehityksessä
- uudet alat, ammatit, työtehtävät yms. perustuvat uusiin ohjelmistoihin, jotka ovat osa uusia, entistä laajempia tieto- ja informaatiojärjestelmiä
- tietoa täytyy hallita ja muokata älykkäästi sekä tarjota sitä jalostuneessa muodossa, jalostuneina palveluina käyttäjien jokapäiväisten rutiinien helpottamiseksi ja päätöksenteon tueksi
 - tarve ja kysyntä laadukkaille ohjelmistoille kasvaa taatusti!
- Huom: Ohjelmistoihin ja niiden kehitykseen liittyviä työ- ja liiketoimintamahdollisuuksia yksittäisestä kehittäjästä, konsultista ja kouluttajasta esim. yli 100.000:n työntekijän jättiyrityksen teknologiajohtajaksi yllin kyllin!
 - Avoin ja vapaa ohjelmistokehitys! (mm. muuttaa (liike)toiminta- ja oppimismalleja ja -mahdollisuuksia...)

2. ohjelmistotekniikasta ja -prosesseista

Ohjelmistotekniikka – mitä se on?

- 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. (kuulostaako tutulta?)
- 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.

- Nykyisin SE määritellään(kin) monella tavalla, joista alla yksi:

"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)

- SE:stä lisää: [Software Engineering 2004 Volume](#), http://en.wikipedia.org/wiki/Software_engineering

SE tänään:

- monien kehitysvaiheiden tuloksena, "jättiläisten 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"
 - yhdellä yksittäisellä tekniikalla ei saada kaikkea aikaan!
- sillan- ja talonrakennus(insinööri)tieteisiin nähden edelleen suuria eroja, erityisesti kvantisoinnissa
 - mutta meillähän kaikki on tietokoneella alusta loppuun?
- (näyttävien) ohjelmien tekeminen liian helppoa – lapsikin sen osaa, mutta se laatu on kyllä jotain ihan muuta: [Boehm: Quantitative evaluation of software quality](#)
- ohjelmistokehitys- ja suoritussympäristön kokonaisvaltainen ymmärtäminen (ihmiset, softat ja raudat) vaativa homma (vrt. tietoturva)
- osatko ohjelmoida ja/vai ohjelmointikieltä ja/vai uusinta sovelluskehitystä ja/vai palvelualustoja ja/vai laadunhallintajärjestelmää 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**
 - ⇒ **prosessimallit!**
- [Mutta, huom:](#) Ohjelmistokehityksen teoriaa ei ole (toisin kuin OS, PL, DS&AL jms. tekniiset jutut), joka tekee homman akateemisesti haastavaksi!

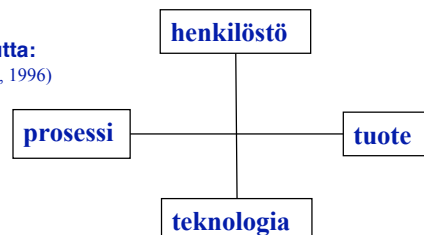
Prosesseista

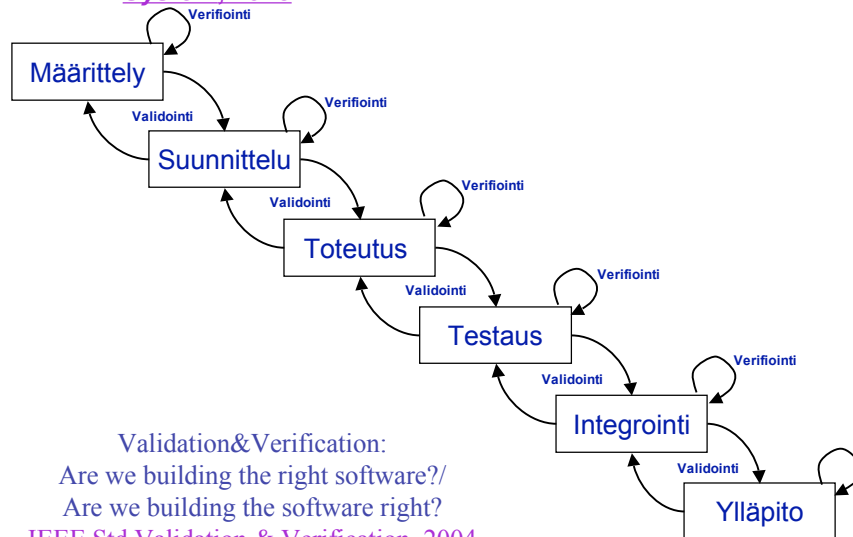
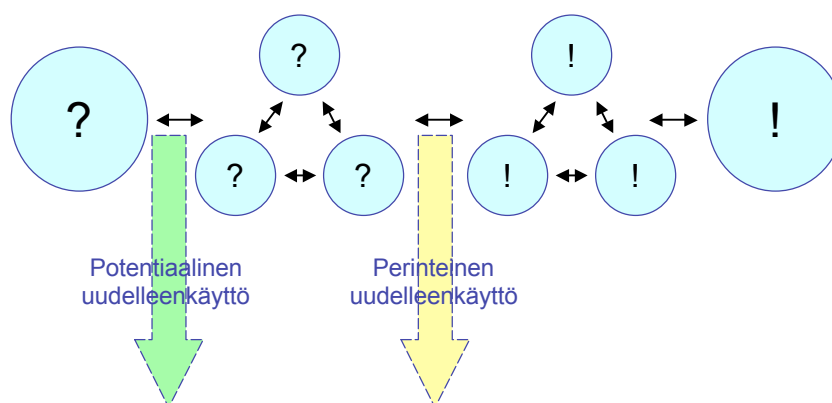
1. Risk-driven approach
 - esim. Boehmin spiraalimalli
2. Document-driven approach
 - esim. UP ja sen variantit
3. Code-driven approach
 - esim. XP
4. Yleisesti:
 - Määritellään kehitysvaiheet ja niiden järjestys
 - a) What shall we do next?
 - b) How long shall we continue to do it?
 - Määritellään vaihetuotteet, inkrementit, prototyypit, validointitekniikat, jne.
 - HUOM: kenen näkökulmasta missäkin vaiheessa suhteessa vaihetuotteisiin?
 - esim. vaatimukset: (liike)toimintaprosessit, asiakasvaatimukset, laadulliset vaatimukset, ...
 - esim. käyttötapausten/käyttäjäkuvausten formaatti
 - käyttäjän/järjestelmän interaktiosekvenssit (*käyttäjä* → *kehittäjä*)
 - käyttötapausten hierarkiat & priorisointi (*käyttäjä* → *projektipäällikkö*)
 - (toiminnalliset) poikkeukset (*kehittäjä* ↔ *käyttäjä*)
 - arkkitehtuuri/laajennettavuus (*kehittäjä* → *projektipäällikkö*)
 - frekvenssi (*kä* → *ke*, *kä* → *propä*, *ke* → *propä*, *propä* → *kä*...)

Muutama pikahuomio prosesseista

- työn jakaminen osiin mahdollistaa työn jakamisen, edistymisen seuraamisen, projektin johtamisen (ja sen osista laskuttamisen) yms.
- laatu (V&V) syntyy osavaiheiden tulona:
 - Vaatimukset 90% oikeita
 - Rakenne 90% oikea
 - Käyttöliittymä 90% järkevä
 - Toteutus 90% kohdallaan
 - Testikattavuus 90%
 - Onnistuminen $0.9 \cdot 0.9 \cdot 0.9 \cdot 0.9 \cdot 0.9 \sim 0.59$ eli 59%:sti!!!
 - iteratiivisuus ja inkrementaalisuus
 - jos tehdään alunperin väärää ohjelmaa, onnistuminen 0%

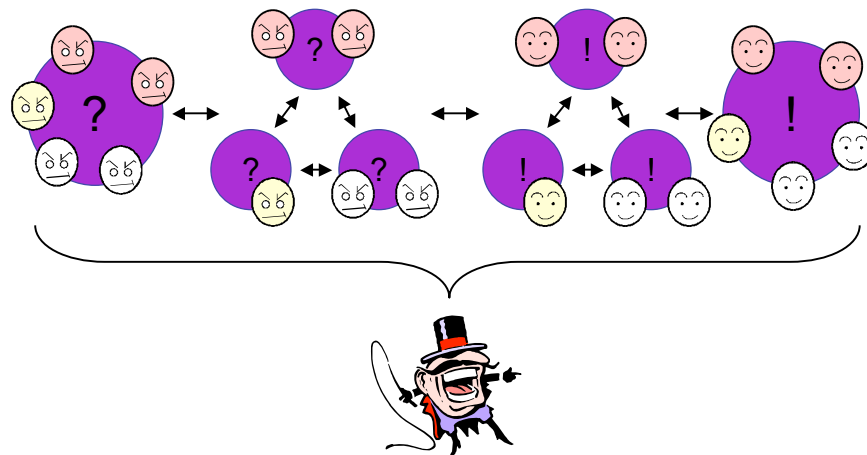
Kehittämisenopeuden neljä ulottuvuutta:
(Steve McConnell: Ohjelmistotuotannon hallinta, 1996)



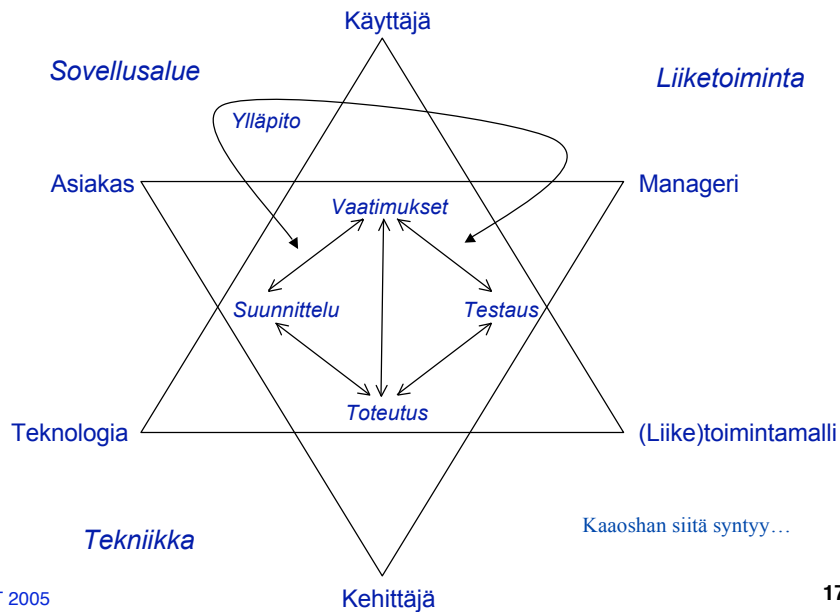
Vesiputousmalli vs Roycen malli, tulkinnat vs. faktat:**Royce: Managing the Development of Large Software System, 1970****Ongelmanratkaisuanalogia:**

1. Mikä ongelma halutaan ratkaista? (vaatimukset)
2. Miten ratkaistava ongelma jakaantuu osaongelmiin? (suunnittelu)
3. Miten osaongelmat ratkaistaan? (toteutus)
4. Mikä saadaan alkuperäisen ongelman ratkaisuksi? (koonti)

Ohjelmistoprojekti



3. Näkökulmia ohjelmistokehitykseen

Rooleja ja näkökulmia**CHAOS Report 1994: Impaired Factors**

Project Impaired Factors	% of Responses
1. Incomplete Requirements	13.1%
2. Lack of User Involvement	12.4%
3. Lack of Resources	10.6%
4. Unrealistic Expectations	9.9%
5. Lack of Executive Support	9.3%
6. Changing Requirements & Specifications	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%
Other	9.9%

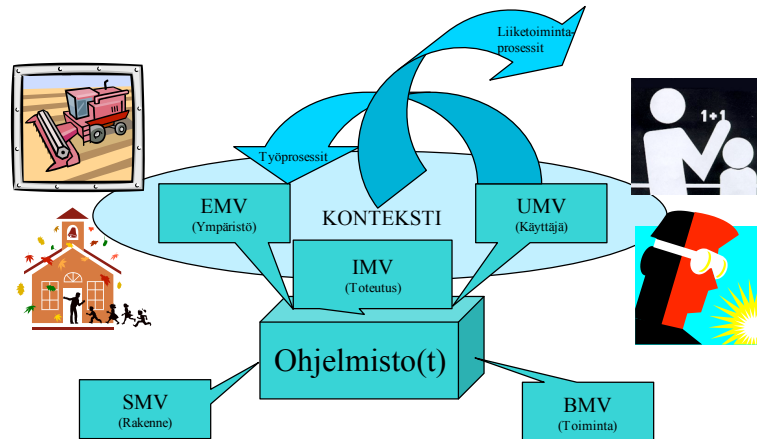
CHAOS Report 1994: Success Factors

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%

Chaos Report 1999

CHAOS Ten: What makes project successful?	
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

Haaste ja mahdollisuus: oikeat ohjelmistot oikeille käyttäjille oikeisiin ympäristöihin!



Paperikoneiden konfauksesta

