

Johdatus ohjelmistotekniikkaan 2003 - siitä se lähtee...

Tommi Kärkkäinen
Jyväskylän yliopisto
tietotekniikan laitos

Agenda

- 1. IT:n kehitys ja merkitys nyt ja tulevaisuudessa**
- 2. Joitakin esimerkkejä alan tutkimuksesta**
- 3. Ohjelmistotekniikan uusi maisteriopintokokonaisuus**
- 4. JOT-kurssin rakenne**
- 5. Ohjelmistokehityksen perusteet**

Miten tähän on tultu? (tietoyhteiskunta)

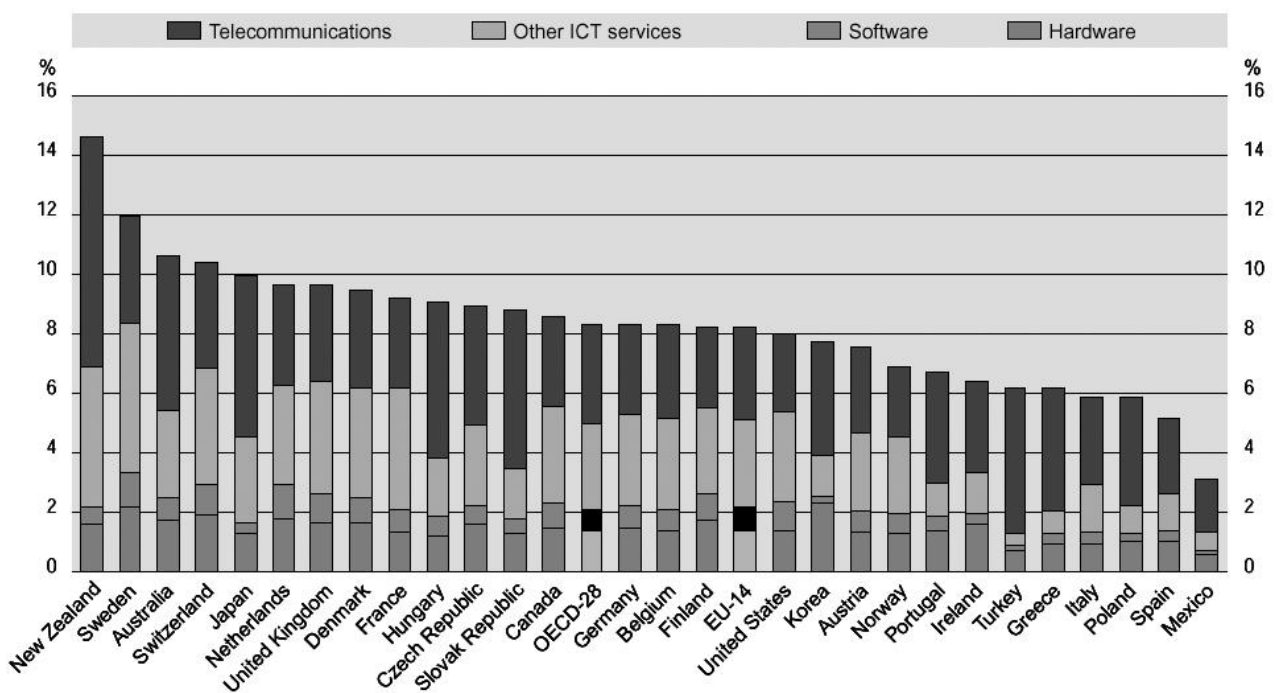
AIKAKAUDET	KERÄILY	MAATALOUS	TEOLLISUUS	TIETO
Perhetyypit	Liikkuva perhe	Suurperhe	Ydinperhe	Hajautunut perhe
Työroolit	Keräilijä	Talonpoika	Asema	Virtuaalitiimi
	Metsästäjä Kaupankäyjä	Paimen Käsityöläinen	Erikoisala Ammattilainen	
Sosiaaliset Merkitykset	Terveydenhoito	Kasti	Yhdistys	Elektroninen ryhmä
	Vapaa-aika Ystävyyys	Luokka Uskonto	Harrastus Kerho, klubi	
Koulutus	Suullinen traditio Autenttinen oppiminen	Kirjallinen traditio: eliitti Yliopisto	Koulu ja luokka: kansanopetus Ammatillinen koulutus: tutkinnot	Oppimisympäristöt Ajan, paikan ja instituution rajat ylittävä oppiminen

- tiedon kuljettaminen kalliimpaa kuin sen liikuttaminen muuten
- aineettoman tiedon (osaamisen) merkitys
- tiedon (ei Internet-tauhkan) tallennus ja uudelleenkäyttö
- digitalisoituminen ja sen tuomat mahdollisuudet (ja haasteet)

(Hannu Linturi, 1997)

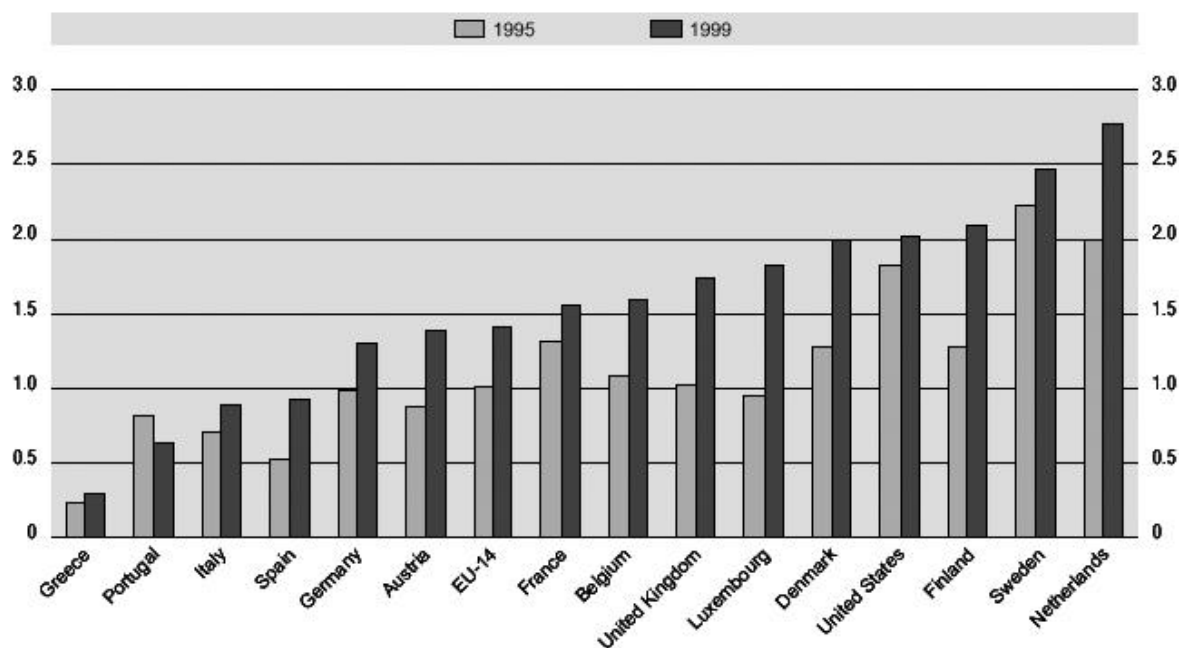
Tietoyhteiskunta/TVT:n merkitys (I)

Figure 27. ICT intensity¹ in OECD countries,² 2001



Tietoyhteiskunta/TVT:n merkitys (II)

Computer workers as a percentage of total employment in selected OECD countries/regions
1995 and 1999



JOT 2003

... nyt ja tulevaisuudessa!

5

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 2003

(OECD Information Technology Outlook – ICTs and the Information Economy, 2002)

6

Miten tähän on tultu? (TIETOTEKNIikka)

TEKNIIKAN MURROS	KÄYTTÖONOTTO SUOMESSA	YLEISTYMINEN SUOMESSA	VAIKUTUS TIETOJEN-KÄSITTELYYN	TIETOJENKÄSITTELYN TAPAHTUMAPAikka
Reikäkorttikoneet				
Reikäkorttikoneiden käyttö tilastointitehtävissä	1923	1920/1930-luvulla	Tietojenkäsittelyn koneellistaminen	Reikäkorttiosasto
Reikäkorttikoneiden käyttö kaupallis-hallinnollisessa tietojenkäsittelyssä	1940-luvulla	1950-luvulla		
Tietokoneet			Keskitetty tietojenkäsittely	
1. ja 2. sukupolven tietokoneet	1958/1960	1960-luvun alkupuolella	Eräkäsittely	Atk-osasto
3. sukupolven tietokoneet	1965	1960-luvun jälkipuolella	Moniajo, suorakäyttö	
Tosiaikaiset päätejärjestelmät	1970-luvun puolivälissä	1970/1980-luvulla	Pääteikäyttö, osituskäsittely, vuorovaikutteinen käsittely	Tehtävien hajautus
Pientietokoneet	1970-luvun puolivälissä	1970-luvun jälkipuolella	Käsittelyn hajautus	Osastokoneet
Mikrosuoritin	1973	1970-luvulla	Toimistoautomaatio	
Mikrotietokone	1970-luvun puolivälissä	1970-luvun lopulla		
PC-mikrotietokone	1983	1980-luvun puolivälissä	Henkilökohtainen tietojenkäsittely	Henkilökohtainen työ-asema, lähiverkot
Internet	1990-luvun puolivälissä	1990-luku	Verkottuminen	Globaalit verkot
Mobiilitieteologia	1990-luvun puolivälissä	1990-luvun jälkipuolella	Paikkasidonaisuuden purkaminen, laitteiden keskinäinen kommunikointi	Lähes missä tahansa

Ketkä näyttävät tietä?

Table A1.6. Top 10 software firms
USD millions and number employed

Company		Revenue 1998	Revenue 2001	Employed 1998	Employed 2001	R&D 1998	R&D 2001	Net income 1998	Net income 2001	Market cap 2001
Microsoft	United States	15 262	25 296	29 159	47 600	2 970	4 380	4 490	7 346	299 900
Oracle	United States	7 144	10 860	36 802	42 927	810	1 139	814	2 561	80 000
SAP	Germany	4 806	5 783	19 308	23 700	597	923	587	585	37 500
Computer Associates	United States	4 206	4 918	11 400	18 200	325	1 110	1 169	-591	16 700
Softbank	Japan	3 922	3 925	1 002	7 219	79	78	11 660
Compuware	United States	1 139	2 010	10 016	13 220	77	116	194	119	3 430
Siebel Systems	United States	418	1 795	1 450	7 389	42	222	8 600
Peoplesoft	United States	1 475	1 737	7 032	8 019	270	329	140	146	7 750
BMC Software	United States	985	1 504	3 604	7 330	164	235	189	42	3 640
Electronic Arts	United States	909	1 322	..	3 500	146	389	73	-11	7 430

Source: OECD, compiled from annual reports and SEC filings.

(huom: mutta minne ja miten? kts. esim. <http://www.sap.com/open>)

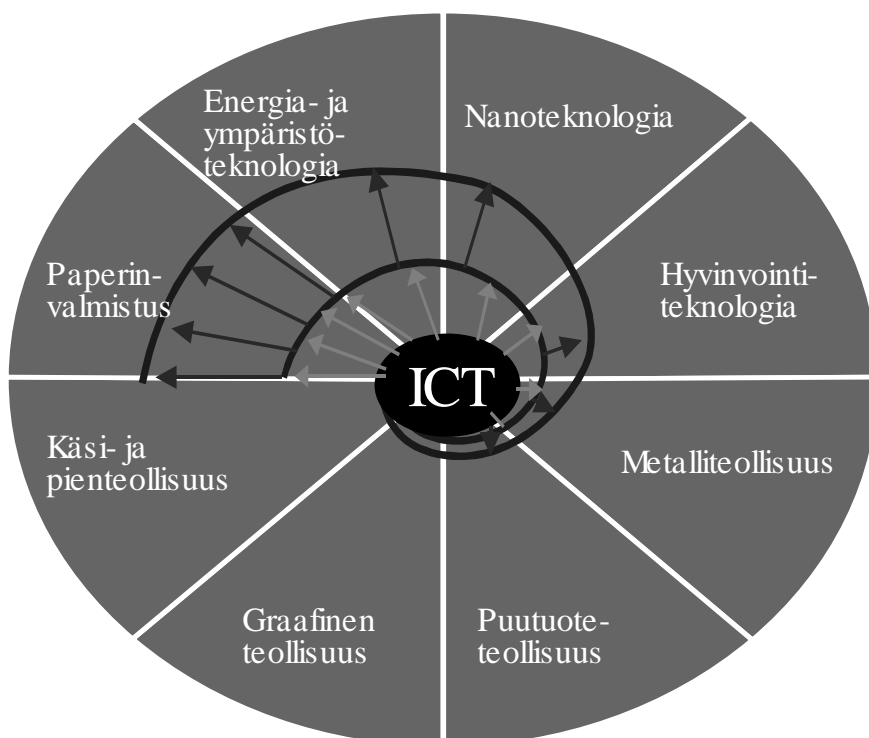
Trendeistä...

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>	Polttoainekennot <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>	Miniatuuri- ja mikroelektronikka <i>(miniaturization)</i>
Modulaari-ohjelmistot <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>		

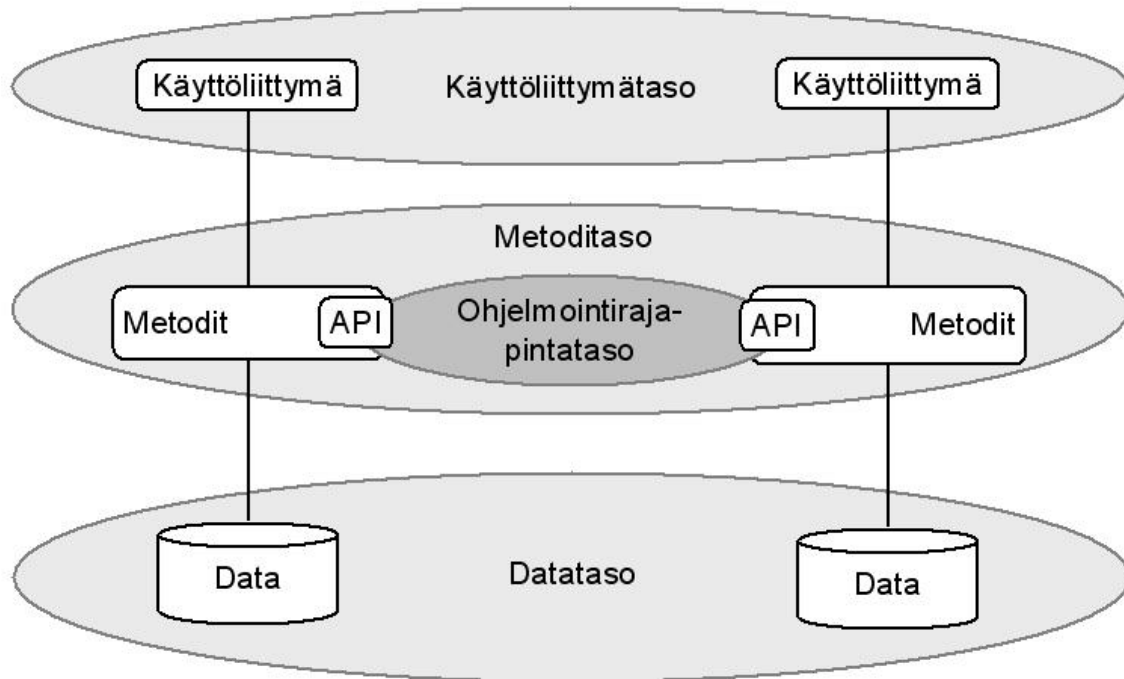
9

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

Tieto- ja viestintäteknologian soveltaminen Keski-Suomen teollisuudessa - alueellinen teknologiastrategia



Tulevaisuuden avainala: integrointi



JOT 2003

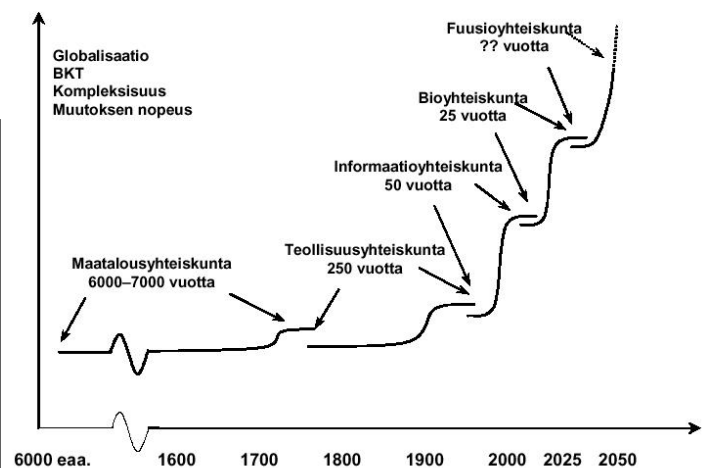
(Esim. Kai Lahti, ohjelmistotekniikan pro gradu -tutkielma, 2003)

11

Trendeistä...

Tulevaisuusammattaja:

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

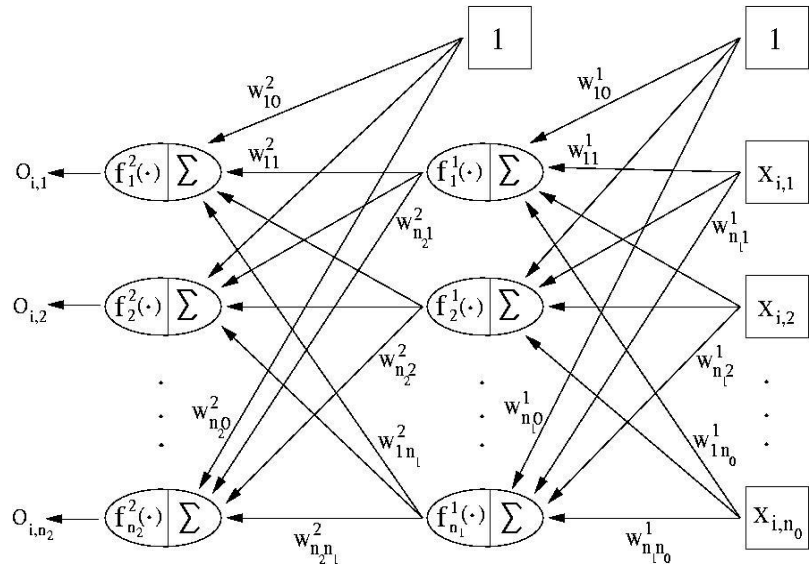
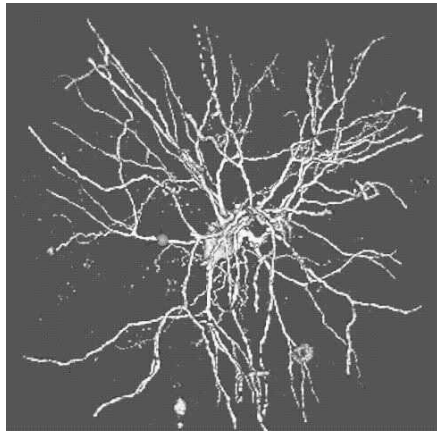


Kuva 2. Yhteiskunnallisen muutoksen suuret aallot (Mannermaa 2002).

→ 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!)

12

Esimerkki: neuroverkko



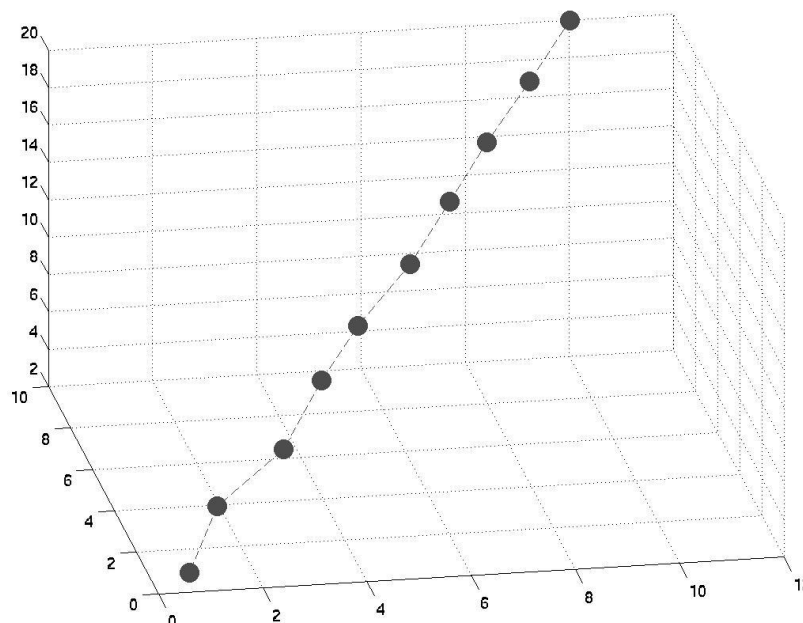
$$y = W_2 \hat{F} (W_1 \hat{x})$$

$$\sum_i |y_i - W_2 \hat{F} (W_1 \hat{x}_i)|^2$$

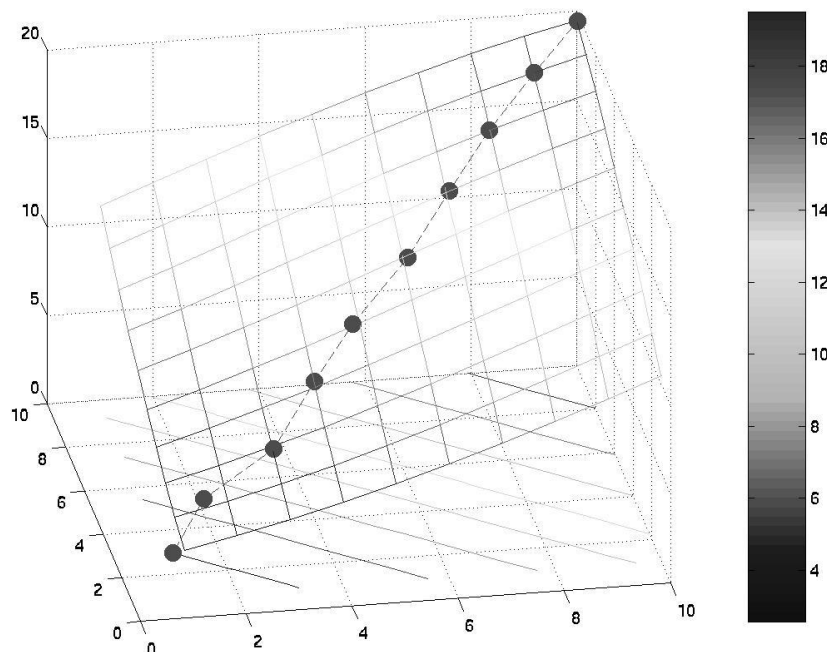
$$\sum_i r(W_1, W_2) = 0$$

↑↑
 Laskennallinen
 malli

Esimerkki: neuroverkon opettaminen (a+b)



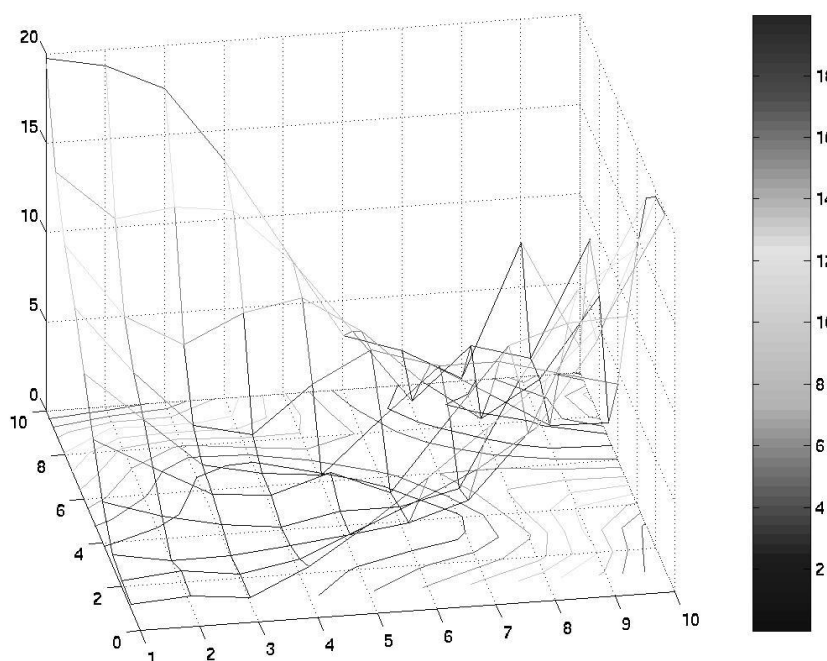
Esimerkki: neuroverkon opettaminen (a+b)



JOT 2003

Saadaan: $(10,1) \rightarrow 10.5, (9,4) \rightarrow 12.9, \dots, (10,10) \rightarrow 19.5, \dots$ 15

Esimerkki: neuroverkon opettaminen (a*b)



JOT 2003

Keskimääräinen virhe 5.5

16

Esimerkki: Nestevirtauksen animointi

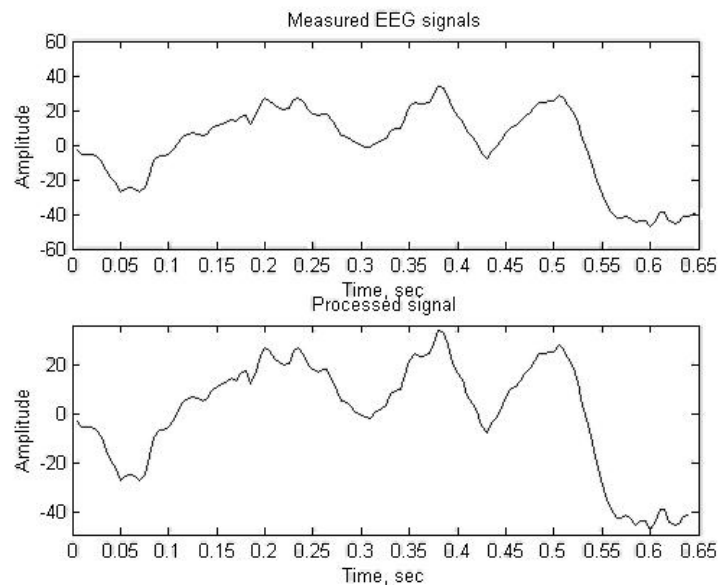
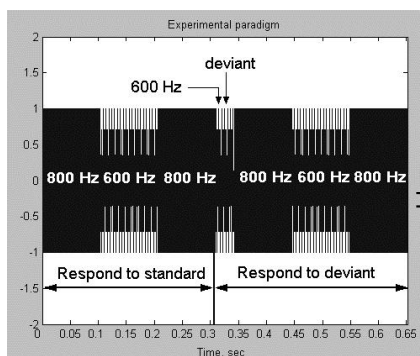


Virtauksen kuvaaminen
Navier-Stokes -yhtälön
avulla

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{F} - \nabla p - (\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u}$$

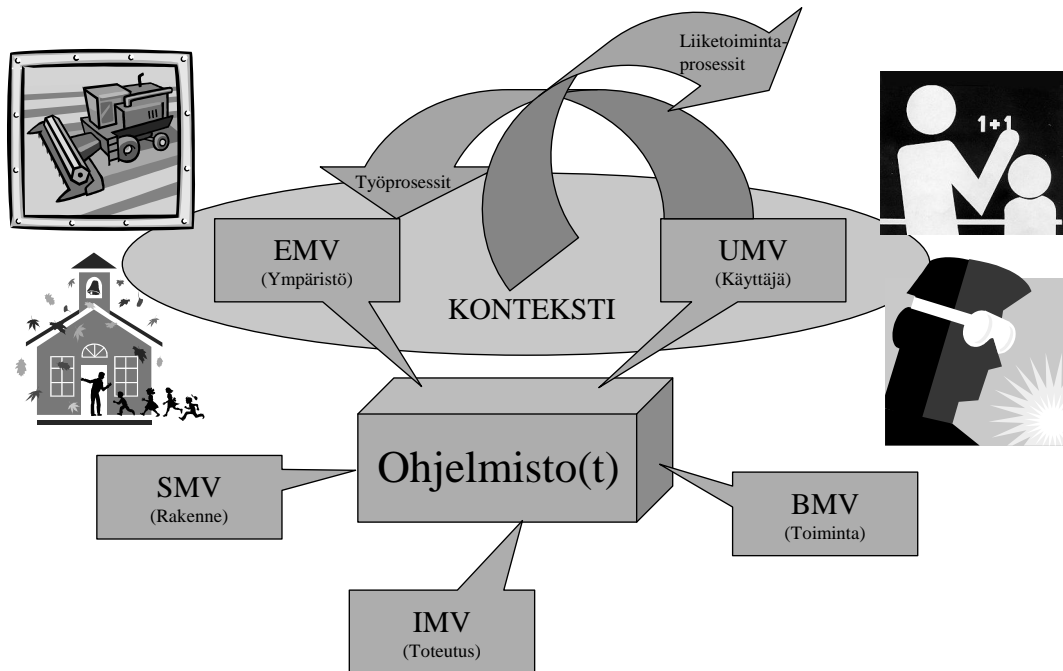
$$\nabla \cdot \mathbf{u} = 0$$

Esimerkki: poikkeavuusnegatiivisuus



- poikkeavuusnegatiivisuus (MMN) kuvaa aivojen kykyä vastata ääniherätteeseen
 - havaitaan (tai ei) tietty aika heräteäänen jälkeen
 - vasteaallon ”suuruuden” uskotaan liittyvän esim. dysleksiariskiin
 - aivosähkösignaalit ”puusta katsoen” täysin satunnaisia
- (huom: tämä on sitä hyvinvointiteknologiaa...)

Haaste ja mahdollisuus: oikeat ohjelmistot oikeille käyttäjille oikeisiin ympäristöihin!



Esimerkki: Data Mining -projekti

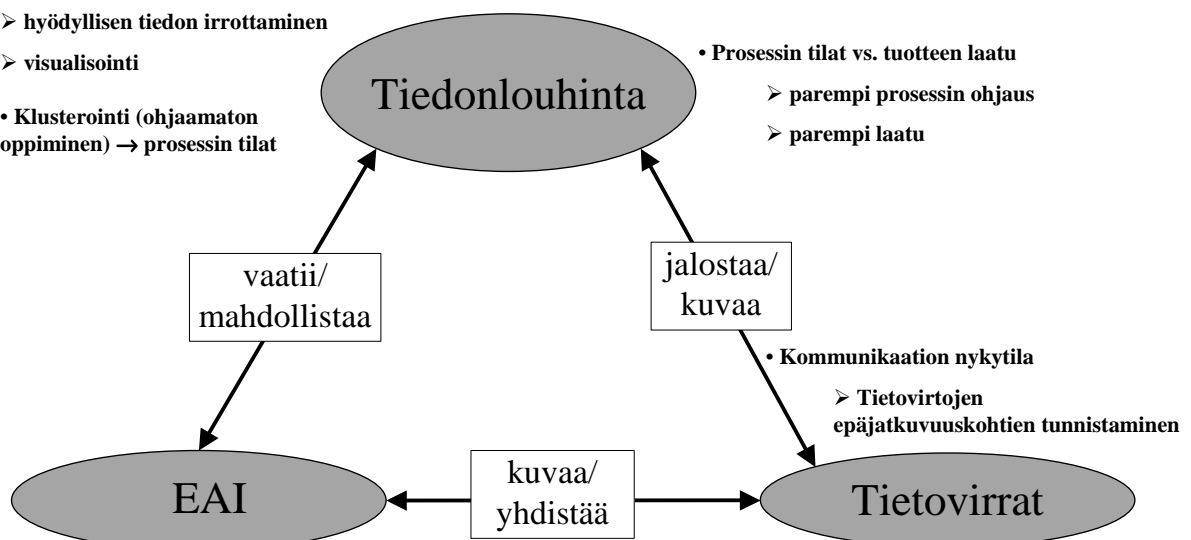
• Suurten tietovarastojen analysointi

- hyödyllisen tiedon irrottaminen
- visualisointi

• Klusterointi (ohjaamaton oppiminen) → prosessin tilat

• Prosessin tilat vs. tuotteen laatu

- parempi prosessin ohjaus
- parempi laatu



• Integrointiarkkitehtuurit ja -tasot

• Väliohjelmistoratkaisut ja suunnittelumallit

• Kommunikaation nykytila

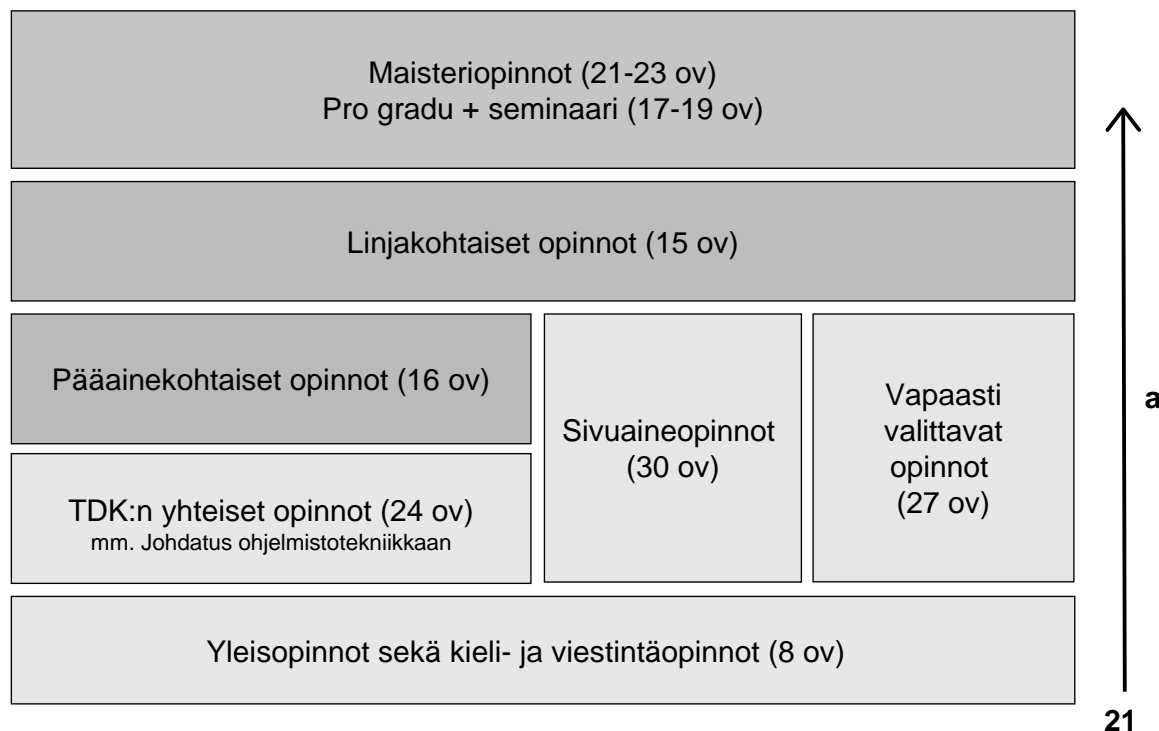
- Tietovirtojen epäjatkuvuuskohtien tunnistaminen

• Kommunikaation kuvaaminen Genrejen avulla

• Analyysi rooleittain

• Konkreettinen lähtökohta liiketoiminnan kehittämiseksi

uusi tutkintorakenne (ohjelmistotekniikka)



Tiedekunnan yhteiset opinnot (I)

- Ohjelmointi 1 (4 ov)
 - Ohjelmoinnin peruskurssi Javalla
- Tietoverkot (2 ov)
 - Yleiskuva tietoliikenteestä ja tietoliikennepalveluista
- Ihminen ja tietojärjestelmät (2 ov)
 - Ihmisen ja tietokoneen välisen vuorovaikutuksen (HCI) perusteet
- Oliokeskeinen tietojärjestelmien kehittäminen (2 ov)
 - Tietojärjestelmien kehittäminen oliolähestymistavan ja erityisesti UML:n avulla

Tiedekunnan yhteiset opinnot (II)

- **Johdatus ohjelmistotekniikkaan (2 ov)**
 - Yleiskäsitys ohjelmistotyöstä
 - Esitiedot: (olio-)ohjelmointia, (olio-)suunnittelua, HCI:tä ja tietoverkkoja
- Tietokannat ja tiedonhallinta (4 ov)
 - Tutustuttaa tietokannan hallintajärjestelmien peruskäsitteisiin, periaatteisiin, arkkitehtuureihin ja kieliin
- Algoritmit 1 (2 ov)
 - Perustietorakenteet ja -algoritmit
- Käyttöjärjestelmät (2 ov)
 - Käyttöjärjestelmän rakenne ja toimintaperiaatteet
- Kandidaatin tutkielma ja seminaari (3+1 ov)
 - Tieteellinen kirjoittaminen, esittäminen ja opponointi

Pääainekohtaiset opinnot (tietotekniikka)

- Ohjelmointi 2 (4 ov)
 - Ohjelmansuunnittelua ja olio-ohjelmointia Javalla
- Algoritmit 2 (2 ov)
 - Algoritmien suunnittelumenetelmiä
- Sovellusprojekti (7-8 ov)
 - Työelämälähtöisen ohjelmistoprojektin toteuttaminen ryhmätyönä
- Harjoittelu (3 ov)
 - Alan työtehtäviin tutustuminen

Ohjelmistotekniikan linjakohtaiset opinnot

- Tietoliikenneprotokollat (2 ov)
 - Tietoliikenneprotokollat ja protokollaohjelmointi
- Automaatit ja kieliopit (3 ov)
 - Äärelliset automaatit ja säännölliset kielet
- Ohjelmistotuotanto (4 ov)
 - Laajojen ohjelmistojen suunnittelun ja toteutuksen menetelmät
- Valinnaisia (6 ov)

Ohjelmistotekniikan maisteriopinnot

- **Ohjelmistotekniikan pakolliset, 12 ov:**
 - ohjelmistoarkkitehtuurit 3 ov
 - testaus ja laadunvarmistus 3 ov
 - kaksi seuraavista:
 - tiedonhallinnan jatkokurssi 3 ov
 - algoritmien teoria 3 ov
 - ohjelmistojen ylläpito 3 ov
 - olio-ohjelmointi 3 ov
- **Ohjelmistotekniikan menetelmät ja periaatteet**
 - ohjelmointikieliä, formaaleja menetelmiä, automaattista päättelyä, funktio-ohjelmointia, ...
- **Adaptiivinen ja älykäs tietojenkäsittely**
 - data-analyysiä, laskennallista älykkyyttä ja neurolaskentaa, tekoälyä, ...
- **Tiedonhallinta**
 - liikkuvaa tietojenkäsittelyä, tapahtumanhallintaa, tietokantoja, ...

JOT-kurssin lukujärjestys:

Aika	Aihe	Pitäjä
ke 15.10. 12-14 (Aud. 3)	Johdantoluento	Kärkkäinen
ti 21.10. 12-14 (Aud. 2)	Johdanto jatkuu	Kollanus
ke 22.10. 12-14 (Aud. 3)	Prosessit ja UML	Kärkkäinen ja Äyrämö
ma 27.10. 12-14 (Aud. 3)	Unified Process (UP)	Heikkilä
ke 29.10. 12-14 (Aud. 3)	UP jatkuu	Heikkilä
ti 4.11. 12-14 (Aud. 1)	Ketterät prosessit, XP	Itkonen
ke 5.11. 12-14 (Aud. 3)	Testaus	Itkonen
ti 11.11. 12-14 (Aud. 2)	Ohjelmointikielen valinta yms.	Kaijanaho
ke 12.11. 12-14 (Aud. 3)	Käyttöliittymän suunnittelu	Kollanus
ti 18.11. 12-14 (Aud. 2)	Tietojärjestelmäarkkitehtuurit	Kollanus
ke 19.11. 12-14 (Aud. 3)	Ylläpito ja tuotteenhallinta	Koskinen
ti 25.11. 12-14 (Aud. 2)	Uudelleenkäyttö ja komponentit	Itkonen
ke 26.11. 12-14 (Aud. 3)	Laadunvarmistus	Heikkilä
ti 2.12. 12-14 (Aud. 3)	Laatujärjestelmät	Heikkilä
ke 3.12. 12-14 (Aud. 3)	Tietoturva	Itkonen
ma 8.12. ???	Agentit yms.	Itkonen

Välitilinpäätös:

- IT/TVT:llä keskeinen rooli yhteiskuntakehityksessä
- uudet alat, ammatit, työtehtävät yms. laskeutuvat uusiksi ohjelmistoiksi, 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!
 - JOTin tarkoitus käydä läpi ydinasiat ohjelmistokehityksestä sekä joitakin "snapshotteja moderneista teknologioista"
 - projektinhallinta, syvempi abstrahointi yms. jätetään ohjelmistotuotannon kurssille
- Huom: ohjelmistokehityksen haasteet eivät ole muuttuneet miksikään kymmeneen vuosiin, ovatpa vain tarkentuneet eri ajanjaksoille ominaisilla piirteillä!

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-historia-artikkelikokoelman <http://citeseer.nj.nec.com/229833.html> 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"

JOT 2003 (IEEE Standard Glossary for Software Engineering Terminology)

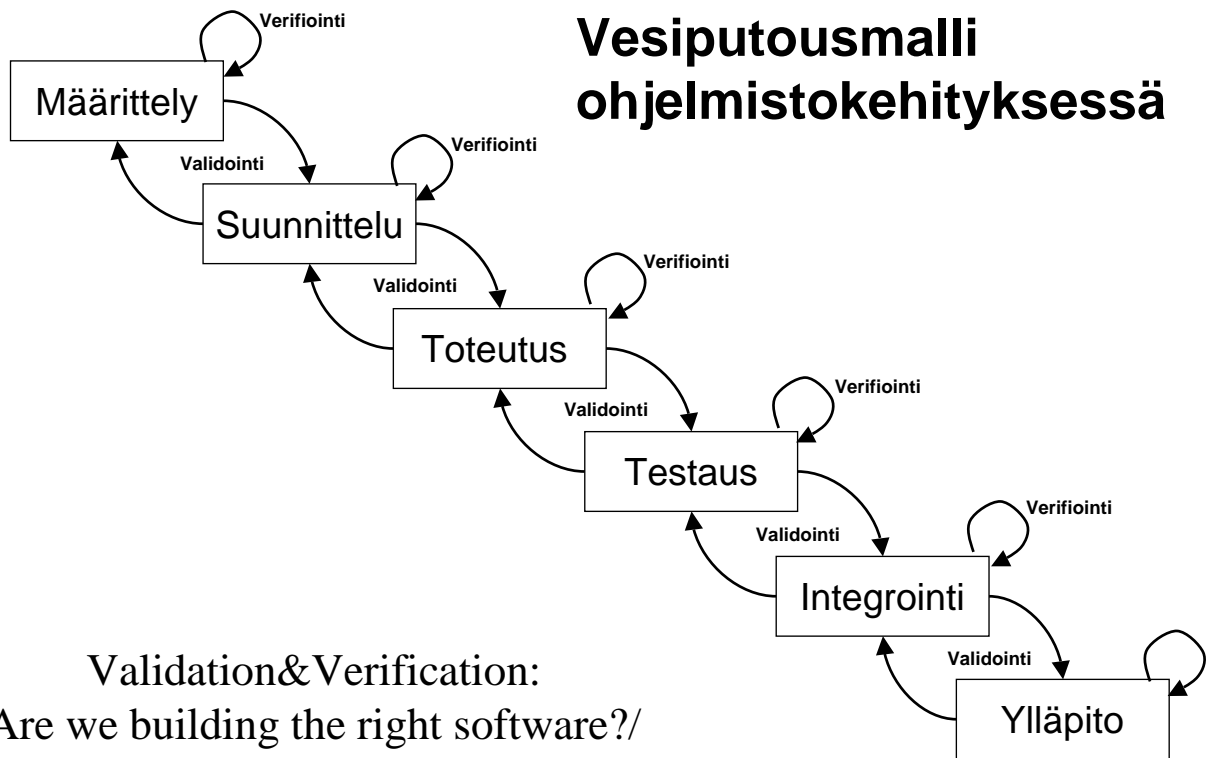
Ohjelmistokehityksen kehitystä:

	1960 ± 5 <i>Programming- any-which-way</i>	1970 ± 5 <i>Programming- in-the-small</i>	1980 ± 5 <i>Programming- in-the-large</i>	1990 ± 5 <i>Programming- in-the-world</i>
<i>Specifications</i>	Mnemonics, precise use of prose	Simple input-output specifications	Systems with complex specifications	Distributed systems with open-ended, evolving specs
<i>Design Emphasis</i>	Emphasis on small programs	Emphasis on algorithms	Emphasis on system structure, management	Emphasis on subsystem interactions
<i>Data</i>	Representing structure, symbolic information	Data structures and types	Long-lived databases	Data & computation independently created, come and go
<i>Control</i>	Elementary understanding of control flow	Programs execute once and terminate	Program systems execute continually	Suites of independent processes cooperate

SE tänään:

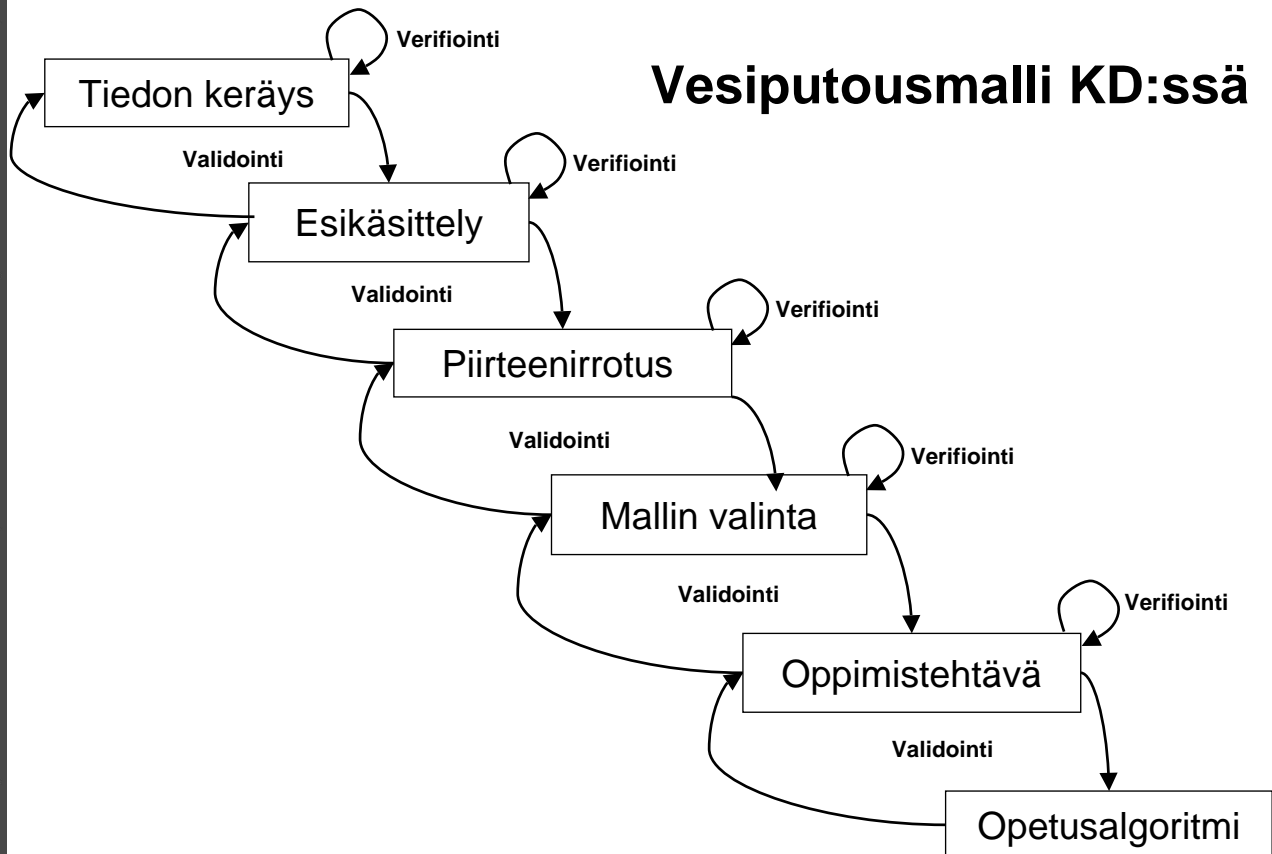
- monien kehitysvaiheiden tuloksena, ”jättiläisten harteilla”;
kts. esim. <http://www.sdm.de/en/it-wissen/themen/sdmkonf-2001/index.html>
- Brooks (1987) ”No silver bullet: Essence and accidents of software engineering”
➤ yhdellä yksittäisellä tekniikalla ei saada kaikkea aikaan!
- sillan- ja talonrakennus(insinööri)tieteisiin nähden edelleen suuria eroja, erityisesti kvantisoinnissa
 - mutta meillähän kaikki on tietokoneella alusta loppuun?
- (näyttävien) ohjelmien tekeminen liian helppoa – lapsikin sen osaa...
- ohjelmistokehitys- ja suoritussympäristön kokonaisvaltainen ymmärtäminen (ihmiset, softat ja raudat) vaativa homma (vrt. tietoturva)
- osatako ohjelmoida ja/vai ohjelmointikieltä ja/vai uusinta editoria?
- osatako sovelluskehittäjiä ja/vai palvelualustoja ja/vai käyttöliittymäeditoreita?
- osatako XML:ää ja/vai RDF:ää ja/vai SOAPia ja/vai...
- Joo, kaikkea pitää osata (tai oikeastaan riittää osata oppia ja tajuta), mutta silti vieläkin tärkeämpää ohjelmistokehityksen suhteen on tietää **kuka tekee, mitä tekee, miksi tekee ja miten tekee!**

Vesiputousmalli ohjelmistokehityksessä



Validation & Verification:
 Are we building the right software?/
 Are we building the software right?

Vesiputousmalli KD:ssä

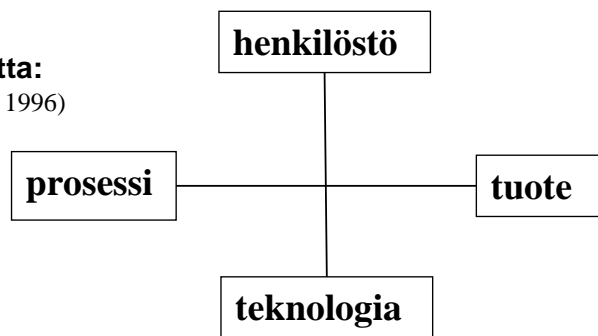


Vaihejakomallin perusominaisuudet:

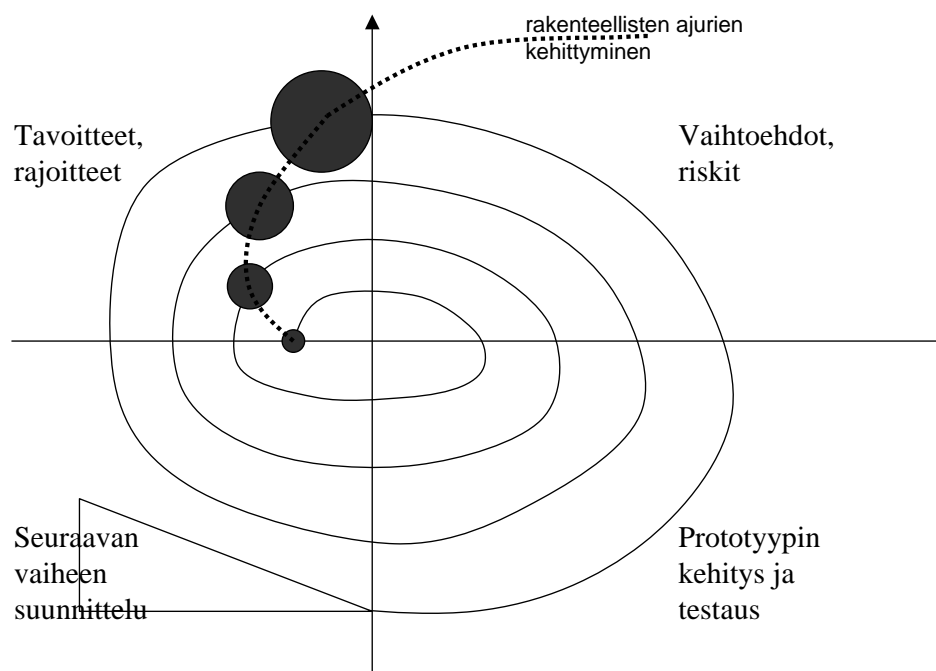
- määritellään vaiheet, aktiviteetit, roolit ja vaihetuotteet
- työn jakaminen osiin mahdollistaa työn jakamisen, edistymisen seuraamisen, projektin johtamisen (ja sen osista laskuttamisen) yms.
- laatu (V&V) syntyy osavaiheiden **tulona**:
 - Vaatimukset 90% oikeita
 - Rakenne 90% oikea
 - Käyttöliittymä 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:

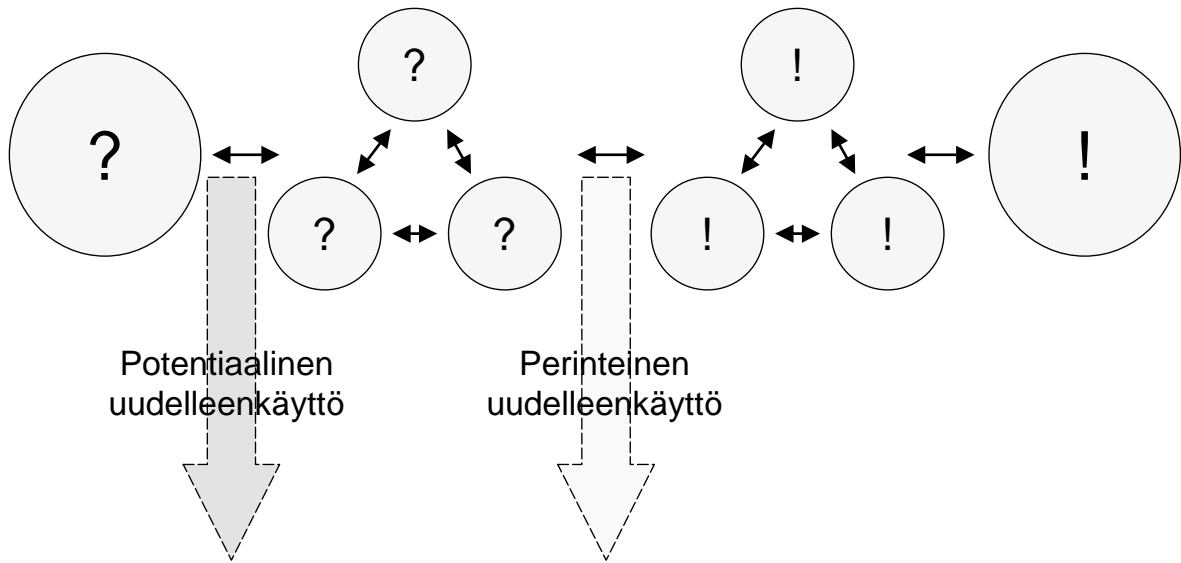
(Steve McConnell: Ohjelmistotuotannon hallinta, 1996)



Spiraalimalli (Böehm)

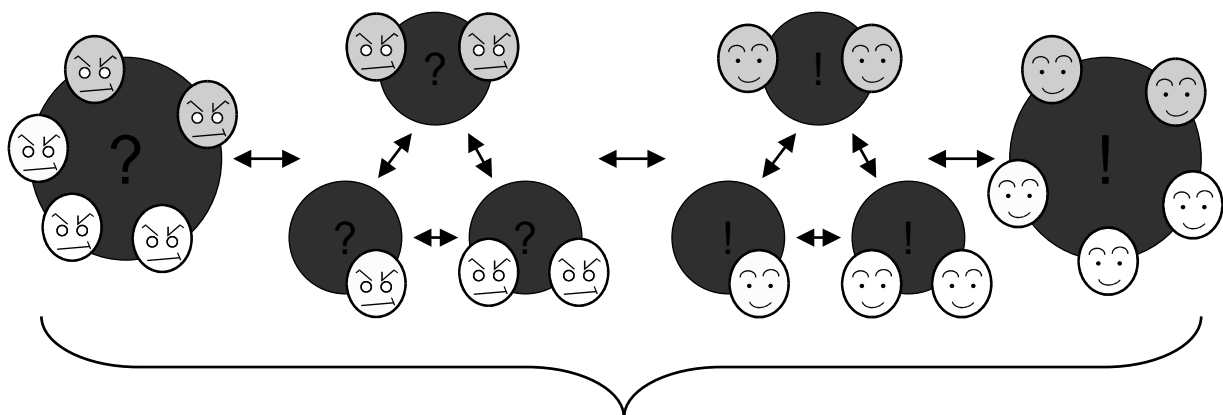


Ongelmanratkaisuanalogia:



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

Ohjelmistoprojekti



Summa summarum - ohjelmistokehityksen kehä JOTin pohjalla:

