

Ohjelmistotuotanto

Johdanto

1

Software engineering

- The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.
(Naur P., Randell B. (eds.): *Software Engineering: A Report on A Conference Sponsored by the NATO Science Committee, NATO, 1968*)

- tekniikkaa
- insinööritaitoa
- kurinalaisuutta
- systemaattisia menetelmiä

© Harri Laine, Jukka Paakk

2

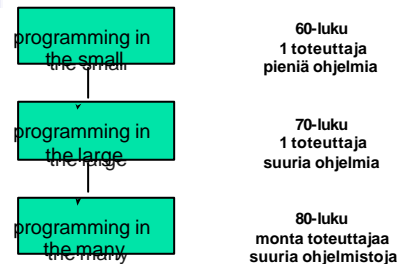
Software engineering

- Käsitteen synty-ympäristö 60-luvun "ohjelmistokriisi" (software crisis) - miten pystytään täyttämään yhä lisääntyvä ohjelmistokysyntä
- Alussa tavoitteita ja toiveita, mutta ei ratkaisuja ongelmiin

© Harri Laine, Jukka Paakk

3

Software engineering - kehitys



© Harri Laine, Jukka Paakk

4

Software engineering - nykyään

- ryhmätyötä
- projektin hallintaa
- integroitua kokonaisuuksia
- sekä teknistä että yleistä osaamista
- kommunikointia
- standardimenetelmiä
- uudelleenkäyttöä
- automaattisia työkaluja
- käyttäjä- ja asiakasläheistä toimintaa
- erikoiskoulutusta
- ylläpitoa

© Harri Laine, Jukka Paakk

5

Software engineering - nykyään

- jatkuvien muutosten hallintaa
- erittäin nopeaa teknologista kehitystä
 - jatkuvaa uudelleen koulutautumista
- kustannusten arviointia
- aikataulujen suunnittelua ja noudattamista

© Harri Laine, Jukka Paakk

6



Ohjelmistotuotannon ongelmia

- Ohjelmistojen koon ja monimutkaisuuden jatkuva kasvu
 - yhä vaikeampia ongelmia ratkaistaan tieto- ja ohjelmistoteknisin keinoin
 - esim. Windows 95: noin 11M riviä koodia
 - esim. Nokian DX200 –puhelinkeskus 12M riviä (30 M riviä (?) räätälöitävää koodia)

© Harri Laine, Jukka Paakkilä

7



Ohjelmistotuotannon ongelmia

- Tuotantoprosessin määrittely ja mittaaminen
 - "mitä ei voi mitata, sitä ei voi hallita"
 - vertailukohteina talon- tai sillanrakennus
 - mitä jos autoja tehtäisiin kuin ohjelmia?
 - liikaa taidetta, liian vähän raakaa työtä
- Tuotteen määrittely
 - asiakas ei tiedä, mitä todella haluaa
 - tieteellisten menetelmien puute

© Harri Laine, Jukka Paakkilä

8



Ohjelmistotuotannon ongelmia

- Tuotteen laadun ja toimivuuden varmistaminen
 - formaali määrittely on vaikeaa
 - täysin kattava testaaminen on mahdotonta
 - ei ole mitään toleransseja
- Jälkitöiden raskaus
 - ylläpitokustannukset ovat usein huomattavasti suuremmat kuin tuotteen kehityskustannukset
- Etukäteisarvioinnin vaikeus
 - projektin vaatiman työmäärän arviointi onnistuu liian harvoin
 - teorian puuttuminen

© Harri Laine, Jukka Paakkilä

9



Ohjelmistotuotannon ongelmia

- Standardiosien puute
 - vain pieni osa (10-20%) kirjoitettavasta koodista on todella uutta, kaikki muu on joskus jossain kirjoitettu
 - ohjelmistokomponenttien kauppa ja kirjastot kehittymätöntä
- Tieteellisen pohjan puuttuminen
 - matematiikka?
 - fysiikka?
 - biologia?
 - humanistiset tieteet?

© Harri Laine, Jukka Paakkilä

10



Ohjelmistotuotannon ongelmia

- Tieteellisen tutkimustyön ja käytännön tuotantotyön välinen kuilu
 - usein: tieteellinen tutkimusongelma \neq käytännön ongelma
 - tieteelliset tulokset usein tarkasti yhteen tiettyyn, pienehköön ongelmaan sopivia
 - yrityksissä ei tunneta tieteellisiä tutkimustuloksia
 - yritysten ratkaisut liian teknologiakeskeisiä
 - yritykset eivät pitkäjänteisesti ratko ohjelmistotuotannon perusongelmia vaan toistavat aikaisempia (huonoja) toimintatapoja

© Harri Laine, Jukka Paakkilä

11



Käsitteitä

Ohjelma

- * tekijänä yksi henkilö
- * henkilökohtaiseen käyttöön
- * kustannukset helppo arvioida

kustannus: 1 yksikköä

Ohjelmatuote

- * laajaan käyttöön
- * yleispätevä
- * dokumentoitu
- * ylläpidettävä
- * ryhmätyn tulos

kustannukset: 5 * n yksikköä

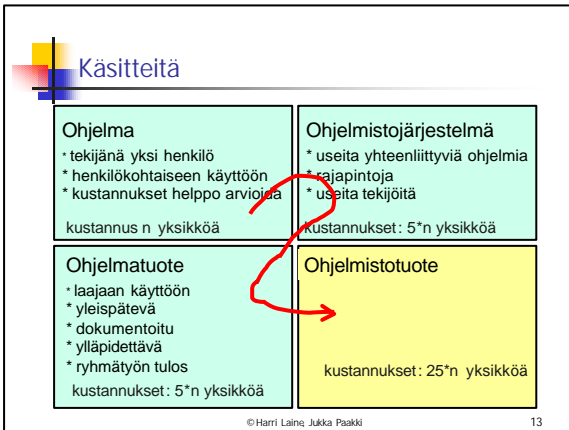
Ohjelmistojärjestelmä

- * useita yhteenliittyviä ohjelmia
- * rajapintoja
- * useita tekijöitä

kustannukset: 5 * n yksikköä

© Harri Laine, Jukka Paakkilä

12



Teollisen ohjelmistotuotannon ongelmia

- ESPITI (European Software Process Improvement Initiative) kyselytutkimukset
- C.Jones: Our Worst Current Development Practices, IEEE Software, March 1996, 102-104
 - Vaatimusten hallinta: mitä on tarkoitus tuottaa?
 - Vaatimusten määrittely yksikäsitteisesti
 - Dokumentointi
 - Testaus ja sen hallinta
 - Projektinhallinta: aikataulut, resurssit, henkilöt
 - Standardien puute
 - Suunnittelun huono laatu
 - Historiallisen mittatiedon vähyys: mitä tuli tehdyksi ja miten?
 - Sitoutuminen epärealistisiin aikatauluihin
 - Riskien hallinta: mitä jos jokin menee pieleen?

© Harri Laine, Jukka Paakkilä 14

ACM CR-luokitus: D.2 SOFTWARE ENGINEERING

- D.2.0 GENERAL
- D.2.1 REQUIREMENTS/SPECIFICATIONS
- D.2.2 TOOLS AND TECHNIQUES
- D.2.3 CODING
- D.2.4 PROGRAM VERIFICATION
- D.2.5 TESTING AND DEBUGGING
- D.2.6 PROGRAMMING ENVIRONMENTS
- D.2.7 DISTRIBUTION AND MAINTENANCE

© Harri Laine, Jukka Paakkilä 15

ACM CR-luokitus: D.2 SOFTWARE ENGINEERING

- D.2.8 METRICS
- D.2.9 MANAGEMENT
- D.2.10 DESIGN
- D.2.M MISCELLANEOUS

© Harri Laine, Jukka Paakkilä 16

