

## Ohjelmistotuotanto

---

### Ohjelmistotuotannon laatu järjestelmät

1

## Ohjelmistotuotannon laatu järjestelmät

- **Laatu järjestelmä:** ohjelmistotuotannon laadun hallinnassa tarvittavat organisaatorakenteet, menettelyt, prosessit ja resurssit
  - määrittelee laadukkaan prosessin menetelmät
  - tarjoaa tavan mitata prosessin ja sitä noudattavien projektien laatua
  - **auditointi:** prosessin / projektin arviointi suhteessa johonkin tiettyyn laatu järjestelmään
  - esimerkkejä: ISO 9000 -sarja, CMM, Bootstrap, SPICE

© Harri Laine, Jukka Paakki 2

## Laatu järjestelmät – CMM

- **CMM (Capability Maturity Model):**
  - W.S. Humphrey: *Managing the Software Process*. Addison-Wesley, 1990
  - alkuperäinen tavoite: USA:n puolustusministeriön käyttämien ohjelmistoalihankkijoiden auditointi
  - kehittäjä: Carnegie-Mellonin yliopiston *Software Engineering Institute (SEI)*
  - joukko ohjelmistoprojekteissa hyväksi havaittuja käytäntöjä jaettuna viiteen "kypsyyss tasoon"
  - ollakseen tietyllä kypsyyss tasolla on organisaation kaikkien projektien noudatettava kaikkia ko. tason (ja sitä alempien tasojen) käytäntöjä ("avainprosesseja")
  - pääpaino prosessissa ja sen hallinnassa, ei tekniikassa

© Harri Laine, Jukka Paakki 3

## CMM-kypsyyss tasot

```

    graph TD
      L1[1. Alustava (initial)] --> L2[2. Toistettava (repeatable)]
      L2 --> L3[3. Määritelty (defined)]
      L3 --> L4[4. Hallittu (managed)]
      L4 --> L5[5. Optimoiva (optimizing)]
  
```

© Harri Laine, Jukka Paakki 4

## CMM-avainprosesit (key process areas)

1. Alustava prosessi
  - ei avainkäytäntöjä
2. Toistettava prosessi
  - vaatimusten hallinta (requirements management)
  - projektin suunnittelu (software project planning)
  - projektin seuranta (software project tracking and oversight)
  - alihankinnan hallinta (software subcontract management)
  - laadunvarmistus (software quality assurance)
  - tuotteenhallinta (software configuration management)

© Harri Laine, Jukka Paakki 5

## CMM-avainprosesit (key process areas)

3. Määritelty prosessi
  - organisaatiokulttuuri (organization process focus)
  - prosessien määrittely (organization process definition)
  - koulutusohjelma (training program)
  - integroitu ohjelmistonhallinta (integrated software management)
  - vaihekohtaiset tuotantotekniikat (software product engineering)
  - ryhmien koordinointi (intergroup coordination)
  - katselmoinnit (peer reviews)

© Harri Laine, Jukka Paakki 6

### CMM-avainprosessit (key process areas)

4. Hallittu prosessi
  - tilastollinen prosessin hallinta (quantitative process management)
  - laadunhallinta ja -mittaaminen (software quality management)
5. Optimoiva prosessi
  - virheiden välttäminen (defect prevention)
  - teknologiamuutosten hallinta (technology change management)
  - prosessimuutosten hallinta (process change management)

© Harri Laine, Jukka Paakki 7

### CMM-avainkäytännöt (key practices)

- Kullekin avainprosessille on määritelty joukko konkreettisia toimenpiteitä
- Projektin seuranta:
  - dokumentoidun projektisuunnitelman käyttö
  - projektin etenemisen seuraaminen suhteessa suunnitelmaan
  - projektisuunnitelman päivittäminen tarvittaessa
  - ulkoisten muutostarpeiden katselointi ja hyväksyminen
  - muutoksista tiedottaminen organisaation johdolle
  - ohjelmiston koon ja laadun seuranta
  - projektin resurssien, kustannusten ja aikataulun seuranta
  - projektin riskien hallinta
  - projektin päätappien ja -tulosten formaali katselointi

© Harri Laine, Jukka Paakki 8

### Ohjelmistotuotanto

Ohjelmistojen tuoteperheet

9

### Tuoteperheet

Tuoteperhe (product family): joukko (ohjelmisto-) tuotteita, joilla on sama perustoiminnallisuus

- tietty sovellusalue (kännykät, sairaalajärjestelmät, ...)
- sama (ohjelmisto)arkkitehtuuri: **yhtenevyys** (commonality)
- tuotekohtaiset erot: **muunneltavuus** (variability)
- yhteinen peruskoodi: **tuoterunko**
- tarjoaa suuren mittakaavan uudelleenkäyttöä yli projektirajojen: vaatimukset, arkkitehtuuri, tuoterunko, komponentit, testiaineisto
- kehittäminen ja testaaminen haasteellisia

© Harri Laine, Jukka Paakki 10

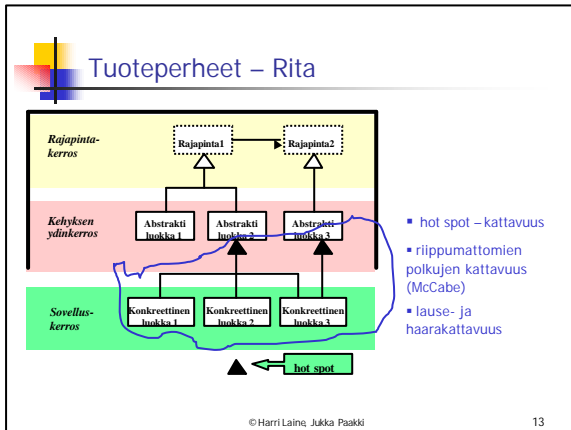
### Tuoteperheet – esimerkki

© Harri Laine, Jukka Paakki 11

### Tuoteperheet – Rita

- Rita (fRamework integration and testing application): tuoteperheiden testaamista tutkiva laitoksen projekti
- tuoteperhe = olioperustainen sovelluskehys
- keskitytään erityisesti sovelluskehiksen ja sovellusten välisen erikoistamisrajapinnan ("hot spots") toimivuuden testaamiseen
- testauksen laatukriteeri: erilaiset kattavuusmitat
- Rita-työkalu tuottaa sovelluskehiksestä (Java-ohjelmasta) vuokaavion, jonka perusteella kattavuus mitataan

© Harri Laine, Jukka Paakki 12



### Ohjelmistotuotanto

Avoimen lähdekoodin ohjelmistokehitys

14

- ### Avoimen lähdekoodin ohjelmistot – Open source software
- Binäärikoodin lisäksi *lähdekoodi* ("source") annetaan ohjelmiston mukana
    - lähdekoodi lukuelpoista ja vapaasti muokattavissa ("open")
    - lähdekoodia saa jaella vapaasti ("free software")
    - erilaisia erityislisenssejä yksityisen kaupallistamisen estämiseksi (GNU General Public License, GPL)
    - muokattu koodi levitettävä edelleen hyötykäyttöön samalla avoimella lisenssillä
- © Harri Laine, Jukka Paakki 15

- ### Open source software
- Juuret 1970- ja 1980-luvulla
    - Internet, Unix
    - Richard Stallman ja GNU (GNU's Not Unix)
    - idealististen hakkeriyhteisöjen synty
  - Kohu 1990-luvulla
    - WWW, Apache
    - Linus Torvalds ja Linux
    - teollisuus peliin: bisnesmallien kokeiluja
    - 1998 *free software* → *open source software*
- © Harri Laine, Jukka Paakki 16

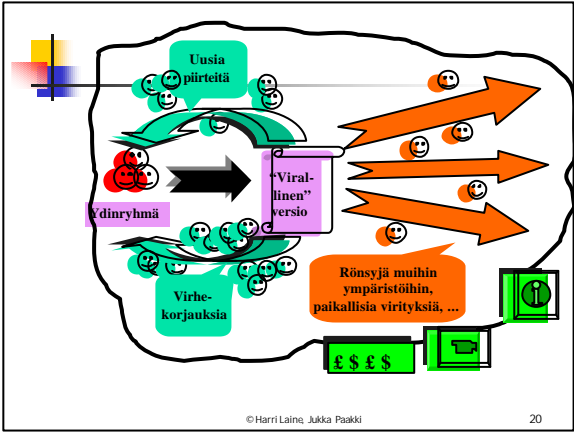
- ### Open source software
- Runsaasti laadukkaita ja halpoja (jopa ilmaisia) ohjelmistokehitysokaluja
    - varusohjelmistot ja käyttöliittymät: Linux, Apache, Netscape/Mozilla, GNOME, KDE, ...
    - ohjelmointi: GNU (emacs, gcc, gdb, ...), Eclipse, Perl, Erlang, ...
    - toimistojärjestelmät: Sun / StarOffice, OpenOffice
    - erikoissovelluksia: WAP-kehitysympäristöt, pelinkehitysalustat, UML-editorit, ...
- © Harri Laine, Jukka Paakki 17

- ### Open source software
- Piilokustannuksia
    - asennuspaketit
    - uusien versioiden käyttöönotto
    - open source -kehitystyön seuraaminen
    - työntekijöiden koulutus
    - (mahdollisesti) virheiden korjaaminen
    - (tarvittaessa) omiin tarpeisiin virittämisen
    - organisaatiossa oltava töissä teknisesti taitavia hakkereita
- © Harri Laine, Jukka Paakki 18

### Open source software

- Avoin ohjelmistonkehitysmalli
  - pienen ryhmän (usein yhden henkilön) kehittämä ohjelmiston runko (*ydin*)
  - ytimen laajentamiseen ja korjailemiseen keskittyvä hajautettu, vapaaehtoinen *hackeriyhteisö*
  - selkeä *ohjelmistoarkkitehtuuri*: ydin + rajapinnat + laajentavat komponentit
  - lähdekoodi* eri versioineen kaikkien vapaasti saatavilla (*Internet*)
  - eri suuntiin rönsyilevä kehitys
  - "virallisen" version kehittymistä valvova *ydinryhmä*

© Harri Laine, Jukka Paakki 19



### Open source software – Linux

- Tutkimus (I.T. Bowman, R.C. Holt, N.V. Brewster: *21st Int. Conference on Software Engineering, 1999*):
  - Linux-ytimen arkkitehtuurin esiin kaivaminen "reverse architecting" -tekniikalla
  - yksinkertaisuus on valttia
  - oppikirjan mukainen perusrakenne
  - mutta**: huomattavasti mutkikkaampi todellinen rakenne
  - mutta**: rajapintoja ei ole tarkasti noudatettu
  - mutta**: tehokkuus >> arkkitehtuurin noudattaminen
- Hakkerit tarvitsevat yhteisen, yksinkertaisen ohjelmistoarkkitehtuurin, mutta eivät jaksakaan tai halua noudattaa sitä loppuun saakka

© Harri Laine, Jukka Paakki 21

### Open source software – Apache

- Apache*: WWW-palvelinten markkinajohtaja (yli 50%, noin 5 miljoonaa asennusta)
- Kehitystyö: alkoi 2/1995, versio 1.0 1/1996, jatkuu
- Tutkimus (A. Mockus, R.T. Fielding, J.D. Herbsleb: *22nd Int. Conference on Software Engineering, 2000*):
  - kehityshistoria 2/1995 - 5/1999
  - 50.000 sähköpostiviestiä, osa muutosehdotuksia
  - versiohistoria
  - 4000 ongelmaraporttia
  - vertailu viiteen "tavalliseen" tuoteprojektiin

© Harri Laine, Jukka Paakki 22

### Open source software – Apache

- Toisin kuin tavallisissa, toimivissa projekteissa**: ei määriteltyä prosessia, ei projektisuunnitelmaa, ei aikataulua, ei tarkkaa teknistä suunnittelua, ei tuotoslistaa, ei dokumentaatiota
- Toisin kuin tavallisissa, hyvin johdetuissa projekteissa**: ei projektipäällikköä, ei tiukkaa kehitystiimiä, ei tehtävänjakoa, ei seurantamekanismeja, ei projekti- tai johtoryhmäkokouksia, ei yhteistä työtilaa vaan hajautettu ja dynaamisesti vaihtuva projektiryhmä
- Kuitenkin**: kehitystyötä koordinoiva ydinryhmä (8-25 jäsentä), joka tekee kaikki tärkeät päätökset

© Harri Laine, Jukka Paakki 23

### Open source software – Apache

- Engelman havaitseminen - vapaaehtoisten etsiminen - (ratkaisu - ratkaisun testaaminen paikallisilla Apache-kopioilla - ydinryhmän suorittama katselointi)\* - koodin vienti kirjastoon ja viralliseen Apache-versioon jakelua varten
- Vaihtoehtoisia ratkaisuja arvioimassa koko yhteisö
- Jokaisella virallisella julkistuksella vastuuhenkilö
- Kehittäjät työskentelevät pääasiassa ainoastaan heille tutun koodin parissa => koodi ei rönsyile liikaa
- Jakeluun hyväksytyillä versioilla 400 ohjelmoijaa
- 460 kehittäjää tuotti muutoksiin johtaneet virheraportit
- 2600 "kehittäjän" raportit eivät johtaneet ohjelmamuutoksiin
- Toisin kuin tavallisissa projekteissa (?): pieni ydinporukka hoiti kehitystyön, valtaosa tuottamattomia vapaamatkustajia

© Harri Laine, Jukka Paakki 24

### Open source software – Apache

- 15 ydingurua tuotti 83% muutosehdotuksista ja 88% koodista
- Yli 3000 hakkeria tuotti virheraportteja eli "osallistui testaukseen" (tosin valtaosin turhaan, ilman muutosvaikutuksia)
- 15 ahkerimmasta virheraporttijaista ydinguruja oli vain 3
- 15 (osa-aikaisen) Apache-gurun keskim. tuottavuus 4300 riviä / vuosi, (päätoimisissa) tuoteprojekteissa 5400-38600 riviä / vuosi
- Vastaava muutosehdotusten käsittelyteho: Apache 110 / vuosi, tuoteprojektit 20-90 / vuosi
- Muutosehdotuksista 50% (ne, joista useimmat käyttäjät olivat riippuvaisia) ratkaistiin ja korjaukset levitettiin 1 päivässä
- Ydinryhmä  $n$  henkeä, korjausryhmä  $10n$  henkeä, testausryhmä  $100n$  henkeä (viimeinen massakin tarvitaan ehdottomasti)
- Kehittäjät ovat myös käyttäjiä

© Harri Laine, Jukka Paakki 25

### Open source software – Apache

- Käyttäjien** raportoima virhettiheys: Apache 2.6 virhettä / 1000 riviä uutta koodia, tuoteprojektit 0.1 - 0.11 virhettä
- Kehittäjien** raportoima virhettiheys (ennen julkistusta): Apache 2.6 virhettä / 1000 riviä uutta koodia, tuoteprojektit 5.7 - 6.9 virhettä
- Apache-projektista puuttuu varsinainen validointi- ja hyväksymistestausvaihe: hakkerit testaavat koodia vain omasta kokeneen käyttäjän näkökulmastaan
- Hakkerit tuottavat rakenteellisesti ("white-box") parempaa koodia ja katselmoivat koodinsa yksityiskohdat paremmin

© Harri Laine, Jukka Paakki 26

### Täyttäkää kurssikysely

<http://www.cs.helsinki.fi/kurssit/kysely/>

© Harri Laine, Jukka Paakki 27

### Lähiajan kokeet

- Kurssikoe: tiistai 18.3, klo 16-20, päärakennuksen sali 1
- Uusintakoe / erilliskoe: perjantai 4.4, klo 14-18, auditorio

© Harri Laine, Jukka Paakki 28