Software Business, Ohjelmistotuotannon laatu & laatujärjestelmät

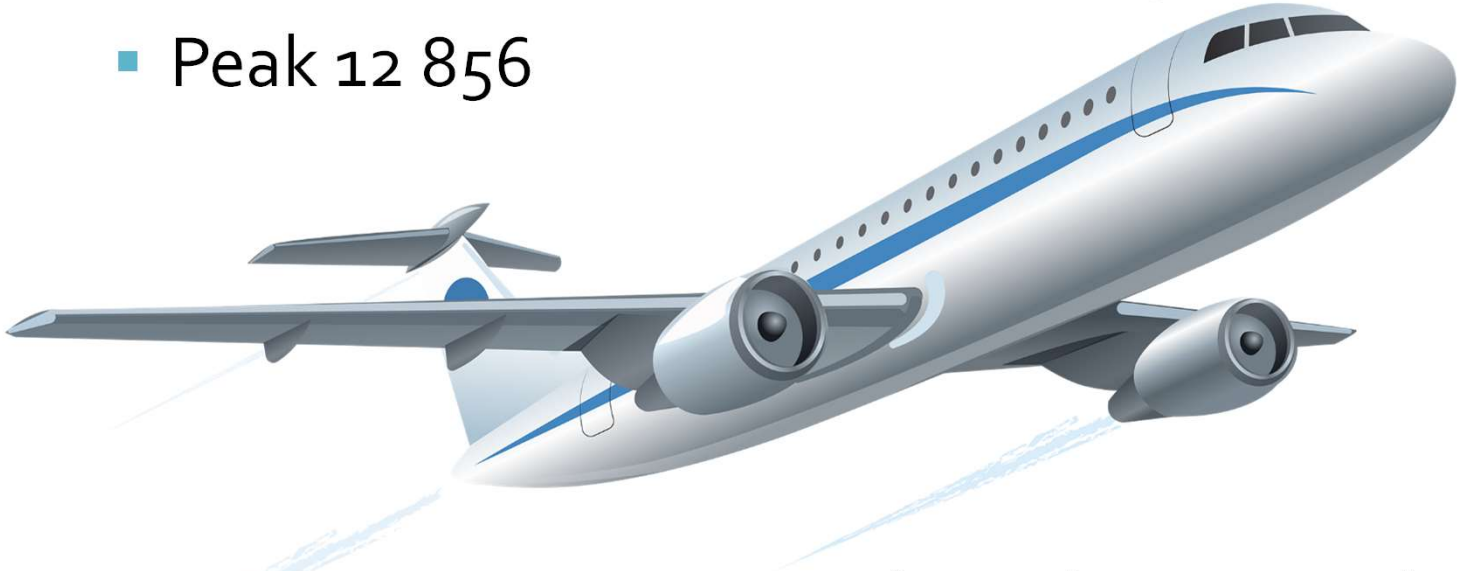
Esa Salmikangas
JAMK/IT
versio 18K

Onko 99% riittävän hyvä?

Onko 99,9% riittävän hyvä?

Miksi 99% ei riitä lentoyhtiöille?

- 9 728 planes in the sky at any given time
  - Peak 12 856



- Jos 1% onnettomuus → **97** konetta alas !
- Jos 0,1% onnettomuus → 9 konetta alas !

Johdantoa ohjelmistotuotannon laatuun

Software bugs… are they any?

- History's worst software bugs by Wired:
  - http://www.wired.com/software/coolapps/news/2005/11/69355
- The top 10 IT disasters of all time
  - 1. Faulty Soviet early warning system nearly causes WWIII (1983)
  - 2. 2. The AT&T network collapse (1990)
  - http://news.zdnet.com/2424-9595_22-177729.html

Case: Explosion of the Ariane 5 On June 4, 1996

- An unmanned Ariane 5 rocket launched by the European Space Agency exploded just forty seconds after lift-off. The rocket was on its first voyage, after a decade of development costing $7 billion.
  The destroyed rocket and its cargo were valued at **$500 million**.
- The cause of the failure was **a software error**:
  - a 64 bit floating point number was converted to a 16 bit signed integer, but the number was larger than the largest integer storeable in a 16 bit signed integer, and thus the conversion failed.
- The error occurred in a software system that was not needed during launch!
  - it was an inappropriate reuse of a 10- year old software component
  - exception handlers had been placed around 4 of 7 variables; unfortunately, the data conversion error occurred in one of the 3 variables, which were left unprotected, since exception handling code makes the system slower

inappropriate *1* sopimaton (epäasiallinen, epäasianmukainen), epätarkoituksenmukainen, tarkoitukseen sopimaton, asiaankuulumaton

Aloitetaan peruskäsitteistä...
"Mikä ohjelmistotuotanto?"

- Defining software engineering -- again!
- You can find many definitions of software engineering. The very first may be the following, from a 1969 conference sponsored by the North Atlantic Treaty Organization (NATO):
  - Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.[2]
- Today, the accepted definition is this one from the Institute of Electrical and Electronics Engineers (IEEE), issued in 1993:
  - Software Engineering: The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.

http://www.ibm.com/developerworks/rational/library/dec05/pollice/index.html

Peruskäsitteet: Ohjelmiston laatu

** IEEE610.12*

Standard Glossary of Software Engineering Terminology

Laatu: (1) Taso, jolla järjestelmä, komponentti tai prosessi täyttää sille määritellyt vaatimukset.

(2) Taso, jolla järjestelmä, komponentti tai prosessi täyttää asiakkaan tai käyttäjän tarpeet tai odotukset.

11.1 Laadun käsitteestä

Lähtökohtia: virheitä esiintyy…

- ohjelmistoissa on vakavia laatuongelmia;

Ohjelmistobugi antoi ilmaista bensaa

Tankkaus on onnistunut ilmaiseksi useilta St1-ketjun asemilta Rovaniemen seudulla. Tapauksien syynä on bensiiniautomaattien ohjelmistovirhe, joka on jättänyt satunnaisesti korttitankkauksia tilittämättä eteenpäin.

Yhtiö joutuu nyt laskuttamaan jo vuonna 2004 tapahtuneita tankkauksia pankki- ja luottokorttiasiakkailtaan. Menovettä tankanneet saavat normaalin tiliveloituksen sijasta laskun kahden vuoden viiveellä kotiinsa.

Lasku on odotettavissa noin 2 000 asiakkaalle. Veloittamatta jäänyt summa kohoaa 180 000 euroon. Yhtiö kertoo tapauksen selvittelyn maksaneen sille tähän mennessä noin 30 000 euroa.

Tapauksen jäljille päästiin, kun yhtiön oman kirjanpidon ja bensa-automaattien raporttien todettiin eroavan toisistaan, kertoo St1 Finlandin toimitusjohtaja **Juha Kokko** STT:lle.

Ketjulla on 270 suomalaisen jakeluaseman lisäksi itsepalveluasemia myös Ruotsissa.**JML**

Tietoviikko 16.6.2006 / Juha-Matti Laurio

Oraclen tietokannassa turva-aukko

Noin kuukausi sitten Oracle 8 -tietokannasta havaittiin ohjelmistovirhe, joka mahdollisti hakkerin tunkeutumisen unix-palvelimeen.

"Aukon kautta voi saada lähes järjestelmän ylläpitäjän oikeudet", sanoo *Dan Sugalski* Oregonin yliopistosta.

Oracle on tiedottanut asiasta vain webin kautta asiakkaille, jotka maksavat tuesta.

"Jos kyseessä olisi ollut suuri riski, olisimme ilmoittaneet asiasta näkyvämmin. Mielestäni teimme kaiken tarpeellisen", perustelee Oraclen palvelinmarkkinoinnista vastaava johtaja *Jeremy Burton*.

Sugalskin mielestä ongelman kokoa kasvattaa se, että tuotteella lienee paljon Linux-käyttäjiä, jotka eivät tiedä tästä asiasta mitään.

Lähtökohtia: haasteita myös…

- kriittiset, sulautetut, kooltaan suuret, reaktiiviset, hajautetut, reaaliaikaiset, aidosti rinnakkaiset
  → ohjelmistot ovat arkipä

Susikoodi

Sampo, Elisa, Finnair, Ake.It-projekteissa syntyy jäljestä päätellen vain sutta ja sekundaa. Vaikka periaatteessa kaikki on niin kovin helppoa.

Teksti: Juha-Matti Mäntylä Kuvitus: Eric Leraillez

On projekteja, jotka menevät hyvin, ja kai niitäkin, jotka valmistuvat etuajassa. Sitten on niitä, jotka eivät mene hyvin eivätkä tyylikkäästi vaan maaliviivan yli ryömien. Tämä on sellainen'', Finnairin Plus-osaston päällikkö **Hanna Kostama** totesi viime syyskuussa.

Finnairin asiakkuusjärjestelmäprojekti takkuili tuolloin pahan kerran. Ensinnäkin se oli viivästynyt vuoden verran, ennen kuin päästiin edes käyttöönottoon. Sitten käyttöönottokatko venyi ilmoitetusta, ja ongelmista raportoitiin vielä yli kuukautta myöhemmin.

Yksittäistapaus? Ei todellakaan.

Viime lokakuussa Tietoviikko kertoi Ajoneuvohallintokeskuksen ikuisuushankkeesta. Vuonna 1999 aloitetun palveluiden ja tietojärjestelmien kokonaisuudistuksen oli alun perin määrä valmistua vuonna 2003, mutta sitä siirrettiin useaan otteeseen. Viime syksynä tähtäin oli jo vuodessa 2011.

Alkuperäinen 16 miljoonan euron kustannusarvio on noussut yli 36 miljoonaan euroon.

Mistä ohjelmiston hyvä "laatu" tulee?

- Ohjelmiston hyvä laatu syntyy ohjelmistoa tehtäessä → sitä ei voi lisätä jälkikäteen eikä käsitellä erillisenä ilmiönä; mielikuva!

Testaaminen taklaa riskejä

Ohjelmistotuotanto törmää liiketoiminnan tavoitteisiin.

Tietoviikko 11.1.2008

Tiina Siltala

Ohjelmistotestaus on aliarvostettu järjestelmäkehityksen osa-alue", Itella Informationin tuotekehitysjohtaja **Pirjo Hannikainen** summaa.

Hänen mukaansa testauksen keskeisin rooli on riskien hallinnassa. Testaus varmistaa tuotteen tai palvelun julkaisuvalmiuden tilanteessa, jossa liiketoiminnan vaatimukset ovat ristiriidassa ohjelmistotuotannon kannalta.

"Liiketoiminnan kolme keskeisintä, oleellisinta ja oikeaa tavoitetta ovat asiakastyytyväisyyden turvaaminen sekä liikevaihdon ja liikevoiton kasvattaminen. Testausta tarvitaan, kun näiden tavoitteiden saavuttaminen synnyttää riskejä ohjelmistokehityksessä", Hannikainen selventää.

Hänen mielestään yksi tyypillisimmistä sudenkuopista järjestelmien rakentamisessa on se, että alussa pidetään liikaa kiirettä ja ollaan ylioptimistisia. Vaikka etukeno on hyvä asia, usein se menee hänen mukaansa överiksi. Hannikainen painottaakin realismia.

Laatu säästää aikaa ja rahaa

Softaprofessori suosittaa käyttäjien mukaanottoa ohjelmistokehitykseen.

Tietoviikko 20.10.2006

Seppo Roponen

Professori **Khaled El Emamin** viesti yrityksille on selkeän yksinkertainen. Tietotekniikkahankkeiden laatuun kannattaa panostaa, sillä laatu tuo mukanaan säästöjä.

Joensuussa viime viikolla järjestetty EuroSPI 2006 -tapahtuma toi Suomeen alan kansainvälisiä asiantuntijoita kuten juuri ohjelmistokehityksen laadun parantamiseen ja seurantaan erikoistuneen El Emamin Ottawan yliopistosta.

Laadun paraneminen näkyy jo nyt hankkeiden onnistumisprosenteissa. Aiemmin vain reilu kolmannes tietotekniikkahankkeista onnistui. Nyt osuus on kohonnut puoleen, mitä ei tosin millään muulla toimialalla pidettäisi edes tyydyttävänä.

Myös globalisaatiolla on kanadalaisprofessorin mielestä ollut selvästi positiivinen vaikutus: kiristynyt kansainvälinen kilpailu on nostanut ohjelmistoyritysten työn laatua.

Laadun ”määritelmä”

- Usein ohjelmistotuotannossa laadulla tarkoitetaan tuotetta, joka on ominaisuuksiltaan **riittävän hyvä** ja **virheetön**.
- ISO-standardit määrittelevät laadun siten, että laadukas tuote **täyttää sille asetetut toiminnalliset ja ei-toiminnalliset vaatimukset**.

Laatu vs virheettömyys

- laatu samaistetaan usein virheettömyyteen
- tuote on virheetön jos se toimii määrittelynsä mukaisesti
- virheettömyys ei kuitenkaan ole riittävä ehto laadulle - miksi?
- virheettömyys ei myöskään ole välttämätön
- laadun edellytys - myös virheitä sisältävä tuote saattaa olla laadukas

”Laatu”

- positiivisesti sävyttynyt käsite
- laatuongelmien hyväksyttävyys on tietoteollisuudelle ominaista
  - – vaihtoehtojen puuttuminen
  - – ei ole totuttu parempaan
  - – miksi kilpailu ei toimi laadun edistäjänä
  - – laaduttomuuden vastapainona voidaan saada joitain ylivertaisia
- ominaisuuksia – kompensaatio
- liiketoimintanäkökulma (lyhyt / pitkä aikaväli)
- mielikuva muuttuu ajan myötä!

Laadun määritelmiä

Deming	Asiakkaalle tärkein on tuote
Crosby	Täyttää vaatimukset
Ishikawa	Asiakastyytyväisyys
Wesselius	Laatu = Objektiivisesti(mitattavat) + subjektiivisesti arvioitavissa (odotukset) + arvioimattomissa (mukautuminen) olevat komponentit
PIMS-tietokanta	Laatu = hintaa lukuunottamatta kaikki muut ostopäätökseen vaikuttavat attribuutit
Webster Dictionary	superiority, excellence, that which belongs to something and makes or helps to make it what it is
Oxford Dictionary	erinomaisuuden aste
ISO	tuotteen tai palvelun kaikki piirteet, joilla tuote tai palvelu täyttää sille asetetut tai oletettavat vaatimukset
kauppamiehen aksiooma	Asiakastyytyväisyys on tärkeintä. Tyytyväiset asiakkaat ostavat uudestaan ja kertovat muille hyvästä tuotteesta. Hyvä tuote tarkoittaa parempaa kuin kilpailijoilla.

Laadun määritelmiä

- *ISO*: täyttää sovitut toiminnalliset ja ei-toiminnalliset (kohtuulliset) vaatimukset
- *Wesselius*:
  - *objektiiviset* laatukomponentit (mitattavissa / laskettavissa olevat)
  - *subjektiiviset* laatukomponentit (odotukset, ei-mitattavissa; tuntemus)
  - *arvioimattomissa olevat*: varautuminen muutoksiin ja virhetilanteisiin; mukautuvuus, joustavuus [hyvät suunnitteluratkaisut]. Ks. ISO 9126; kohta 11.5.

Tehtävä: Ohjelmistotekniikan erityispiirteet

- Kuvaa/kerro ohjelmistotekniikan ja muitten tekniikan alojen (merkittävimmät) erot eli mikä erottaa ohjelmistotekniikan muista tekniikan aloista…
- *In english*: Describe the differences between software engineering and other engineering disciplines.

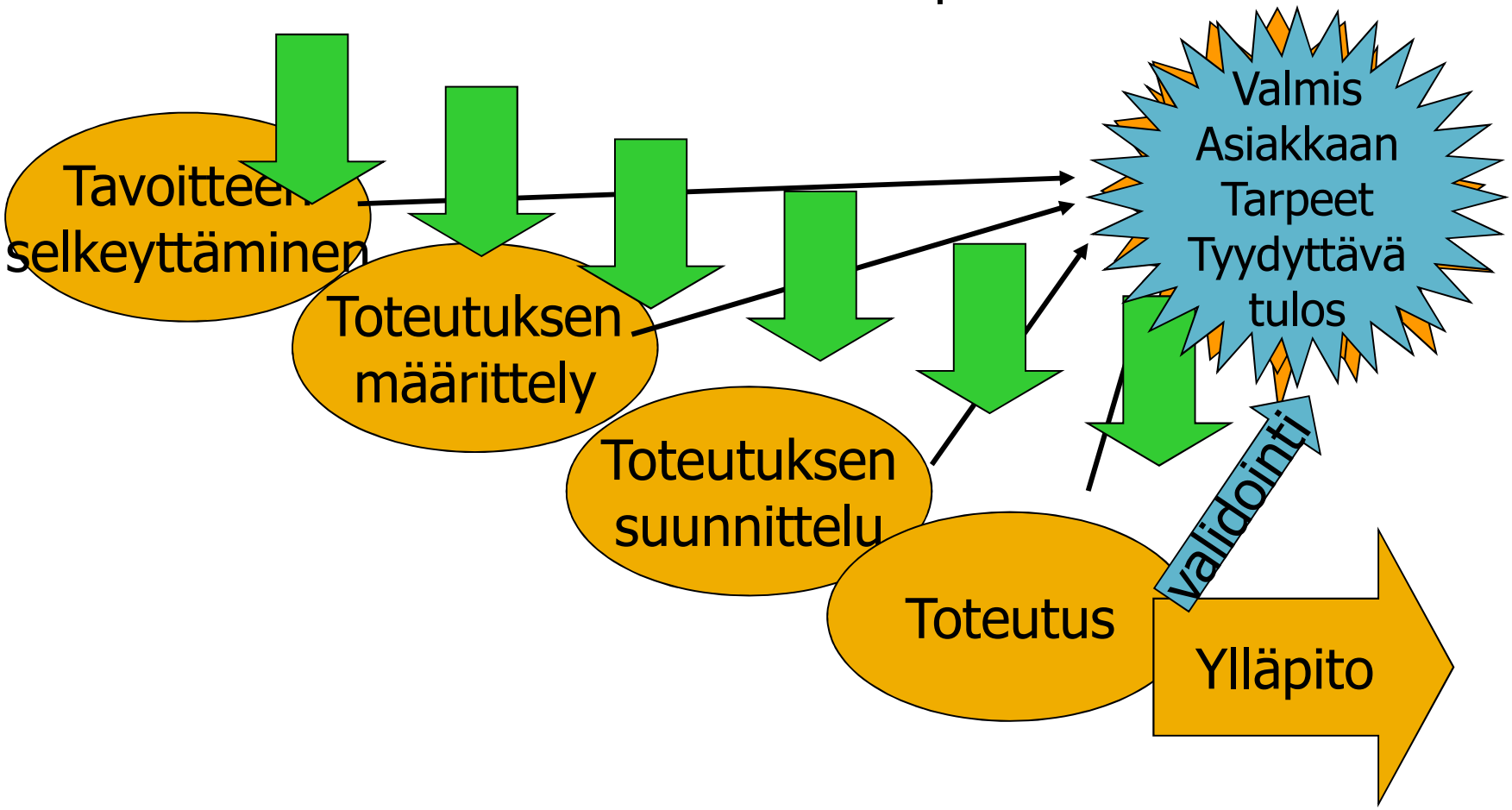
1 discipline ['dɪsɪplɪn] *s*
1 [itse]kuri, järjestys *The teacher was unable to maintain ~.* Opettaja ei saanut pidettyä kuria. *I don't have enough [self] ~ to save money.* Minulla ei ole tarpeeksi itsekuria säästääkseni rahaa.
2 kuritus, kurinpito *He believes in strict ~.* Hän uskoo ankaraan kurinpitoon.
3 harjoitus, harjoittelu *the ~ of meditation* mietiskelyn harjoittelu
4 opinala, tieteenala *When did sociology emerge as a distinct ~?* Milloin sosiologiasta tuli erillinen tieteenala?

Tehtävän: vastaus

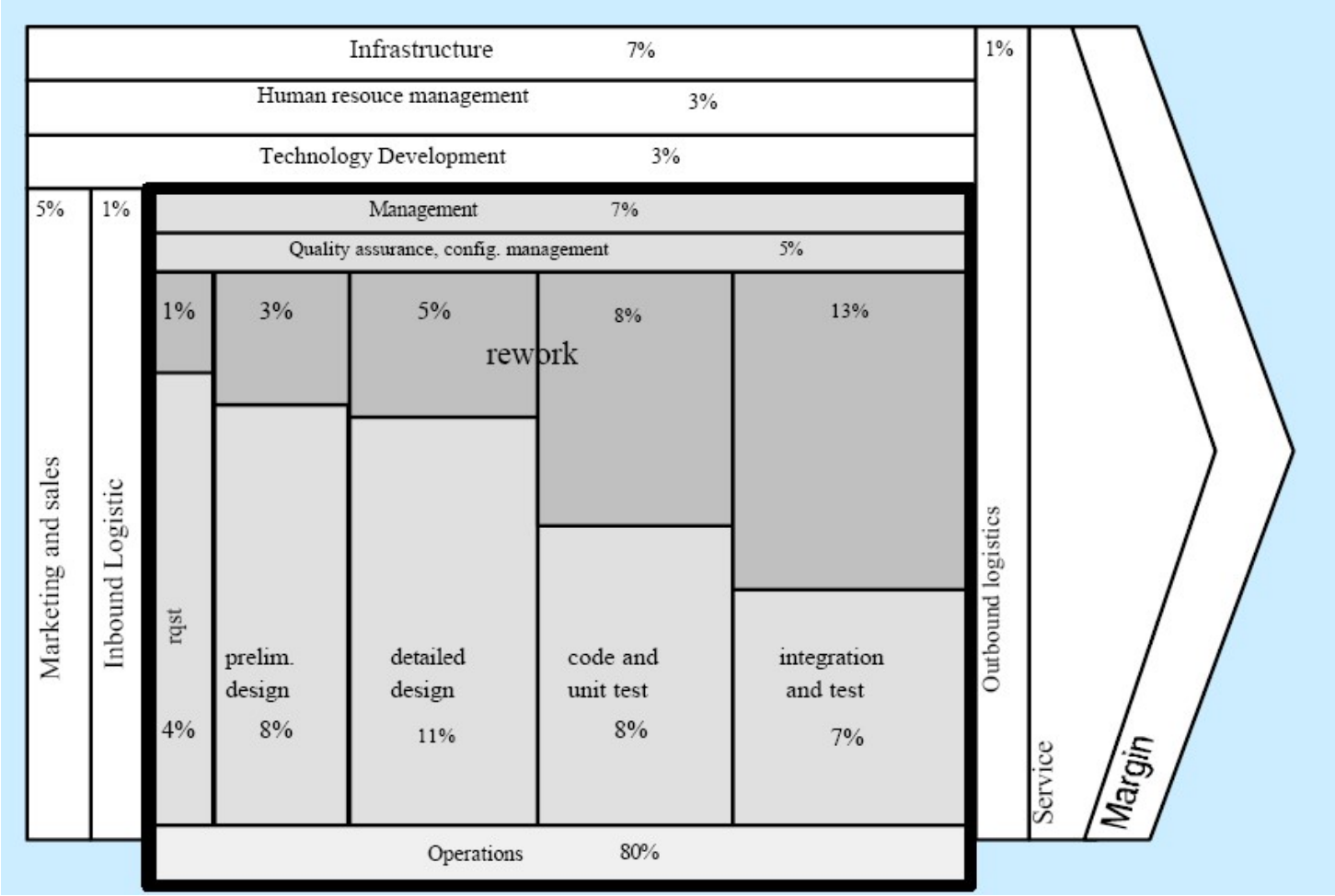
- Ei-fyysinen tuote ~ abstraktius
- Monistaminen (ja jakaminen) helppoa
- Helpompi varastaa
- Siedetään enemmän virheitä
- Tuotanto voidaan hajauttaa, ei sidottu paikkaan
- Liiketoimintamallit voivat olla erilaisia
- Tuotannon tekijät erilaiset, henkilö ja hlötyö korostuu
- Hyödynnetään muilla toimialoilla

"The Big Picture of the quality in software engineering"

Erilaiset laatuun tähtäävät toimenpiteet



Laatu "maksaa", sen puuttuminen vielä enemmän → Boehmin arvoketju



Laatu? Mistä liikkeelle?

- Kirjaston sähköiset aineistot:
  - http://www.jamk.fi/kirjasto/tiedonhaku/ekirjasto/anni

Laadunhallintakeinot

- tuotteen laatu
  - verifiointi (todentaminen) ja validiointi (kelpoistaminen)
  - testaus ! suunnitelmallista
  - katselmoinnit, tarkastukset, arvioinnit
- työn laatu
  - osaamisen kehittäminen
- tekemisen laatu (prosessit, työkalut, menetelmät)
  - laatujärjestelmä
  - osaamisen kehittäminen
- johtamisen laatu
  - laatujärjestelmä
  - osaamisen ja johtamisen kehittäminen
  - mahdollistaa laadukkaan toimintakulttuurin

Laadunvarmistus

- Laadunvarmistus ohjelmistotuotantoprojektissa määrittelee **menettelytavat** ja **toiminnot**, joilla pyritään varmistamaan tuotteen ja prosessin laatu.
- Laadunvarmistus kohdistuu siis itse tuotteen lisäksi koko tuotantotoiminnan laatuun.
  - Tuotteen laadunvarmistuksessa pyritään estämään virheiden pääsy tuotteeseen.
  - Työn laatu syntyy osaamisen kehittämisen myötä
  - Toiminnan/tekemisen laadunvarmistuksessa arvioidaan toimintaa (prosessit, työkalut, menetelmät) sekä kehitetään parempia mahdollisia puutteita korjaavia toimintamalleja.

Case Vuitton

...ntä eroa on Louis Vuittonin "Speedy 30" -laukulla ja Alepan muovikassilla ?

- Laukku ja kassi? Molemmilla siis voi kantaa tavaraa, mitäs tästä nyt melua pitämään? Asia alkoi askarruttaa siinä vaiheessa, kun vertasin hintaa:
  - Speedy 30 maksaa satoja euroja ja Alepan muovikassin saa 15 sentillä.
  - Mistä on kyse? Tätä täytyy selvittää.

Case Vuitton

...lmistoalalla kun ollaan, niin tarkastellaanpa ...asiaa tyypillisten ohjelmistovaatimusten näkökulmasta.

- functional specification, performance
- Availibility
- Security
- Reliability
- Scalable
- User Interface
- System Interface

Case Vuitton

Miksi siis toinen maksaa satoja euroja ja toinen 15 senttiä?

- … osittainen vastaus piilee "suunnittelun" tavoitteessa tai koska suomenkielessä ei ole oikein hyvää sanaa termille "design" niin käytetään termiä "design" paremman puutteessa → Mitä tämä design oikein on ja mitä se tarkoittaa?
- Designilla ja sen yhteydessä tuotteeseen tai palveluun on aina jokin tavoite:
  - designin ja tuotteen tavoite Speedy 30:lle ja muovikassille ovat varsin erilaiset.

Design on toiminnallisen ja emotionaalisen kokemuksen suunnittelua valittuun tarkoitukseen valitulle kohderyhmälle.

  - → Siis lähtökohtaisesti nämä kaksi "ohjelmistovaatimuksiltaan" varsin samankaltaiset ratkaisut ovatkin suunniteltu (design) aivan eri tarkoituksiin , toinen on tavarankuljetukseen ja toinen elämyksen tuottamiseen.
    - Tavoitteet ovat erilaiset ja niin myös hinta. Jos designin tekijä onnistuu tavoitteessaan hän myös saa pyytämänsä hinnan.

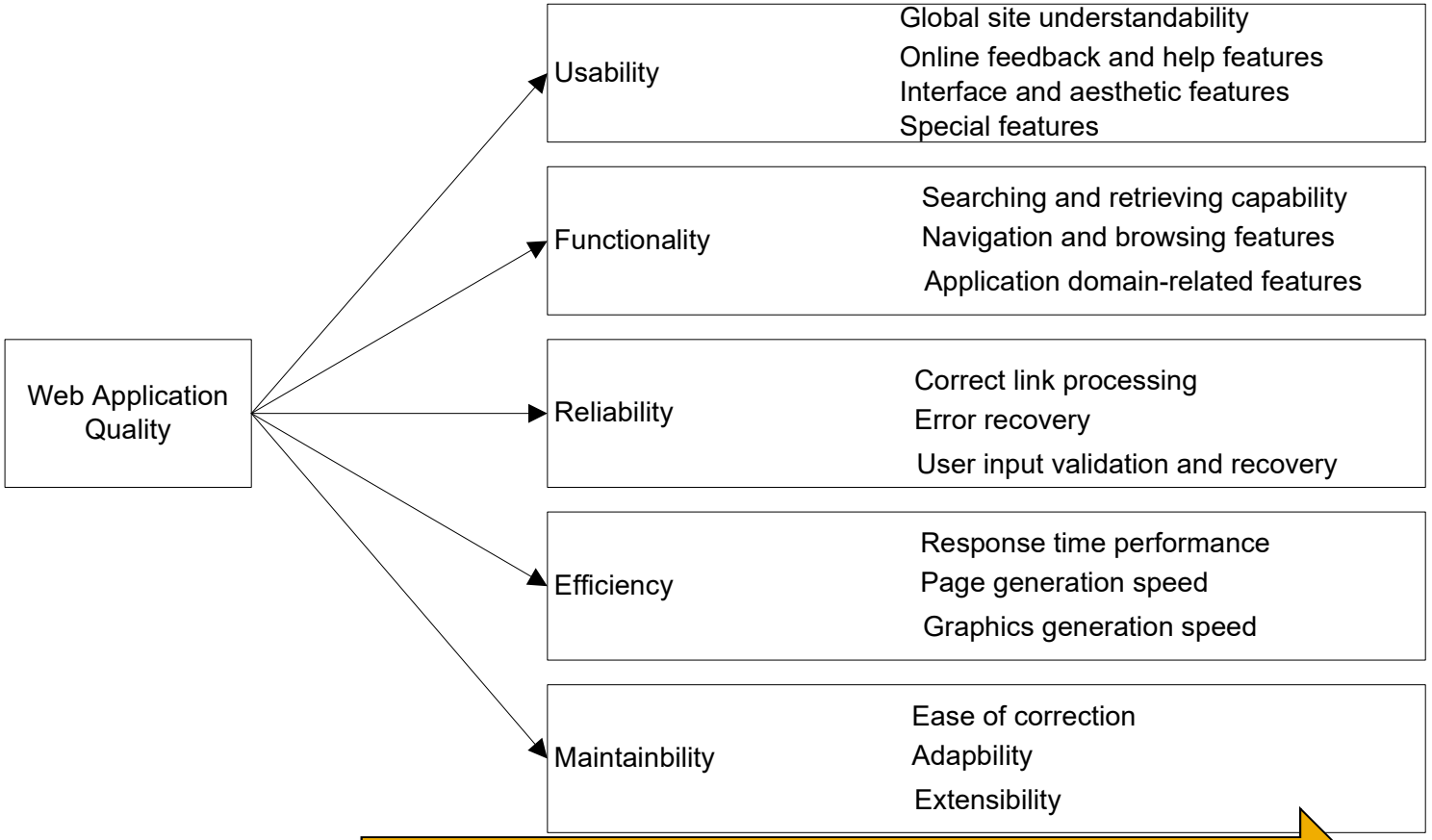
Tehtävä:
Sovellusten laatu

1. Mieti ja dokumentoi mistä muodostuu web-sovelluksen laatu eli mitkä asiat tekevät web-sovelluksesta laadukkaan
   - Mitä erilaisia laadun osatekijöitä pitäisi web-sovelluksen suunnittelussa/toteutuksessa huomioida?
2. Mitkä näistä laadun osatekijöistä ovat yleistettävissä ”mihin tahansa” sovelluksiin?

Tehtävän vastaus:

Quality requirements tree

Olsina et al ”Building web-based…” 1998



Yleisesti ohjelmien laadun osatekijät

Ohjelmiston laatukriteerit

- Ohjelmistotuotteen laatukriteereinä pidetään mm:
  - toiminnallisuutta
  - luotettavuutta
  - käytettävyyttä
  - tehokkuutta
  - ylläpidettävyyttä
  - siirrettävyyttä.

Mieti mikä erottaa/yhdistää termejä:
-Vakiintuneet toimintatavat
-hyväksi havaitut menetelmät
-toiminnanohjausjärjestelmä
- tuotannonohjausjärjestelmä
- laatujärjestelmä

11.2 Laatujärjestelmät

Miksi laatua/laatujärjestelmiä?

- Ohjelmistovirheet maksavat miljardeja kansantaloudelle:
  - ” USA:n kauppaministeriön kansallisen teknologia- ja standardointi-instituutin NIST:n mukaan ohjelmistobugit ja -virheet maksavat maan taloudelle noin **59,5 miljardia dollaria** vuodessa.”
- Lähde: http://m.digitoday.fi/?page=showSingleNews&newsID=20026986


I'M PUTTING YOU IN CHARGE OF GETTING ISO 14000 CERTIFICATION.
S Adams E-mail: SCOTTADAMS@AOL.COM
WHAT'S THE DIFFERENCE BETWEEN THAT AND ISO 9000?
OH, ABOUT 6000. HA HA HA HA!!
9/24/97 © 1997 United Feature Syndicate, Inc.
HEY, I THINK I'LL USE THAT ONE AT THE STOCKHOLDER MEETING.
YEAH, THAT'LL WAKE THEM UP.

Laatujärjestelmän määritelmä

- ISO 8402 standardin mukaan:
  - ”laadunhallinnassa tarvittavien organisaatiorakenteiden, menettelyjen, prosessien ja resurssien muodostama järjestelmä”

Ryhmätyö vk 38/2009

- mieti mikä erottaa/yhdistää termejä:
  - ”tapa toimia”
  - vakiintuneet toimintatavat
  - hyväksi havaitut menetelmät
  - toiminnanohjausjärjestelmä / tuotannonohjausjärjestelmä
  - laatujärjestelmä
- Etsikää webistä lähteitä, joilla ”perustelette” vastauksenne
  - Kirjatkaa lähteet niin kuin JAMKin opinnäytetyöohje määrää
  - ainut lähde jota ei saa käyttää on tämä kalvosarja
- Vastaukset ryhmittäin wikiin
  - Valmistautukaa esittelemään vastauksenne ensi viikolla

Laatu ja laatujärjestelmä #1

- ohjelmiston laatu = tuotteen kyky täyttää käyttäjänsä "kohtuulliset" toiveet ja odotukset
  - subjektiivinen käsite
  - Laatu on aina subjektiivinen käsite → onko standardit "laatua"
- laatu ei tarkoita huippulaatua, vaan erilaisia mitattavia ominaisuuksia
- tuotteen laatuun vaikutetaan parhaiten **toiminnan** laadun kautta
  - Laatutoimenpiteet kohdistuvat toiminnan laadun parantamiseen ja sitä kautta lopputuotteen laadun parantamiseen
- laatujärjestelmä = yrityksen toimintatapa tuotteen, palvelun tekemiseksi/toteuttamiseksi
  - laatukäsikirja, ohjeistukset
  - ohjelmistoyritysten laatujärjestelmät ovat yleensä väljiä

Laatu ja laatujärjestelmä #2

- suunniteltua laatua aikataulun ja budjetin mukaisesti
  - ei huippulaatua, riittävän hyvää laatua
- todistettavuus ja jäljitettävyys
  - ulkopuolisen tahon tai sisäisen laadunvarmistuksen kyettävä varmistamaan että yritys toimii laatujärjestelmän mukaisesti
  - todisteita: tarkastuspöytäkirjat, virheraportit…
- laatujärjestelmän sertifiointi (ISO 9001)
  - määrittelee yleisellä tasolla perusasiat jotka laatujärjestelmän tulee sisältää

Laatu ja laatujärjestelmä #3

- sertifiointi ei todista laatujärjestelmän erinomaisuutta
- käytännössä sertifiointi vaaditaan (asiakaspaineet)

Laatujärjestelmän "kehittyminen"

Kukin tekee niin kuin parhaaksi näkee

homma toimii / ei toimi

analysointi & valinta

Vakiintuneet toimintatavat

toimintatapojen hyvyyden analysointi

hyväksi havaitut menetelmät

Kattavuus

toimintatapojen kehittäminen ja muutosten vaikutusten analysointi

kirjatut prosessit&toimintatavat, seuranta noudatetaanko

ulkopuolinen arviointi

toiminnanohjausjärjestelmä (tuotannonohjausjärjestelmä)

laatujärjestelmä

Sertifioitu laatujärjestelmä

Laatujärjestelmän minimivaatimukset

- laatukäsikirja
- johto sitoutunut
- laatupäällikkö nimetty
- laatujärjestelmä olemassa todistettavasti: löytyy dokumentaatiota
- org. jäsenten toimenkuvat määritelty (tehtävät ja vastuut)
- auditoinnit suunnitelmallisia
- alihankkijoiden toiminta valvonnassa
- dokumenttien hallinta kunnossa
- korjaavat toimenpiteet laatupoikkeamiin kirjattu

Laatujärjestelmän osa-alueet

- laatupolitiikka
  - organisaation laatutavoitteet ja laatupäämäärät
- laatusuunnittelu
  - laatutavoitteiden määrittely
  - laadunvarmistustoimenpiteiden suunnittelu
  - laadunparantamisen toimenpiteiden suunnittelu
- laadunohjaus
  - tekniikat ja aktiviteetit joiden avulla täytetään laadun vaatimukset
- laadunvarmistus
  - sisäinen ja ulkoinen
  - suunnitellut ja systemaattiset aktiviteetit, joiden avulla voidaan todentaa että järjestelmä täyttää laatuvaatimukset
  - testaus, katselmoinnit, tarkistukset
- laadunparantaminen
  - aktiviteettien ja prosessien tehokkuutta ja hyötyä pyritään parantamaan
- laadunmittaaminen
  - tulokset pitää pystyä osoittamaan!
- auditointi
  - sisäinen ja ulkoinen

Ryhmätyö vk 39/2009

1. Mitä eri sertifioituja laatujärjestelmiä on olemassa ”maailmalla”? Nimi, lyhyt kuvaus, lähteet !?
2. Mitä laatujärjestelmiä suomalaiset (teollisuus)yritykset käyttävät? Lähde?
3. Ohjelmistoyrityksillä on tiettyjä eroavaisuuksia tuotannolliseen teollisuutten (mitä?). Mitkä laatujärjestelmät soveltuisivat ohjelmistoyrityksille ja mitkä ei. Perustele.
4. Miksi keski-suomalaisen ohjelmistofirman kannattaisi käyttää sertifioitua laatujärjestelmää? Näkökulmia puolesta ja vastaan.

http://student.labranet.jamk.fi/wiki/index.php/Iizl4010

Siis miksi laatujärjestelmiä...

Source: David Reo, ESI

Näkyvyyttä projektin aikaiseen suoritukseen

Lisää tulosten ennustettavuutta

Parantaa projektin suorituskykyä

Lisää tuotteen laatua

Lisää kyvykkyyttä hallita monimutkaisia projekteja

parantaa työn mielekkyyttä

"Ylimmän johdon organisaatiolle muodollisesti määrittämä yleinen tapa suhtautua laatuun"

11.3 Laatupolitiikka

Systemaattisen ja suunnitelmallisen toiminnan edut

- Organisaation asiantuntija- ja muiden resurssien kohdennus tärkeimpiin asioihin
- Perusorganisaation (usein keinotekoiset rajat) rajat poistuvat
- Systemaattisuus ja suunnitelmallisuus tuovat tehoa
- Vaihtelua rutiineihin
- Mahdollisuus vaikuttaa ja osallistua
- Tehostaa delegoivaa työtapaa
- Tehostaa kommunikointia

"nykyaikaisessa laatujohtamisessa lähdetään siitä että laadukas prosessi tuottaa laadukkaita lopputuotteita"

11.4 Prosessiajattelu
Tuotantoprosessin laatu

Mikä on prosessi?

- Prosessilla voidaan tarkoittaa monia eri asioita:
  - Samanlaisten tapahtumien sarja
  - Yksittäinen tapahtumien sarja (kuten projekti)
  - Tapa tehdä asioita (kuten menetelmä)
  - Joukko aktiviteetteja, ihmisiä, ja lopputuloksia, jotka tarvitaan tuloksen aikaansaamiseksi
- Yleisesti
  - Yhteenkuuluva joukko askeleita, joiden tavoitteena on saavuttaa tietty päämäärä

Eksplisiittinen/implisiittinen

- Prosesseja on kaikkialla – kaikki vaan eivät ole eksplisiittisiä
- Eksplisiittinen prosessi
  - Prosessi, joka on kuvattu eli dokumentoitu
  - Dokumentointi on tehty sovitulla formalismilla ja visualisointitekniikoilla.
- Implisiittinen prosessi
  - Toimintatapa jota ei ole dokumentoitu prosessimallin muotoon
  - Esimerkiksi
    - CASE työkalun mukanaan tuoma toimintatapa
    - Konfiguraation hallinnan dokumentoimattomat toimintatavat

Prosessimallin sisältö

- prosessimalli sisältää seuraavia entiteettejä ja niiden suhteiden kuvauksia:
  - Syötteet ja tulosteet
  - Aktiviteetit
  - Aktiviteetteihin liittyvät henkilöt ja roolit
  - Aktiviteettien ja syötteiden/tulosteiden väliset suhteet
  - Roolien väliset suhteet (kommunikointiverkosto)
  - Teknologiat, tekniikat ja menetelmät
  - Käytetyt työkalut
  - Tuotteiden väliset suhteet (tuotehierarkiat)

Miksi prosessimalleja

- Ymmärrys ja kommunikointi
- Helpottaa projektien suunnittelua ja hallintaa:
  - Mahdollistaa keskittymisen oleelliseen, ei tarvitse keksiä miten ja mitä seuraavaksi pitäisi tehdä!
  - Dokumentoi hyviä käytäntöjä
  - Helpottaa uusien ihmisten sisäänajossa
- Edesauttaa ohjelmiston uudelleenkäyttöä
- Mahdollistaa prosessin hallinnan ja parantamisen (prosessit näkyviksi)

Ohjelmistoprosessin määritelmä

- IEEE-STD-610
  - A sequence of steps performed for a given purpose
- Paulk:
  - A set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products

Tyypillisiä ohjelmistotuotannon prosesseja

- Yleisiä aktiviteetteja ovat esim:
  - määrittely,
  - suunnittelu,
  - toteutus,
  - testaus ja
  - Ylläpito
- **Prosessimalli** kuvaa prosessien organisoinnin

Ohjelmistotuotannon prosessimalleja eli elinkaarimalleja

- "Koodaa ja korjaa"
- Vesiputousmalli (Royce 1970)
- Inkrementaalinen malli (Mills et. Al., 1980)
- Evolutionäärinen malli (Lehman, 1985; Gilb, 1988)
- Prototypointimalli (Curtis et al., 1987)
- Spiraalimalli (Boehm, 1988)
- V-malli (GMOD, 1992)
- "Komponenttipohjaiset mallit" (Mykkänen? 2003)
- Ketterät prosessit)(Agile manifesto 2001)
  - (Agile Processes, esim. XP eXtreme Programming

Laadunvarmistuksen tarkoitus

- Ohjelmisto on käyttäjän kannalta laadukas jos:
  - se vastaa käyttäjän tarpeita
  - on virheetön ja luotettava
  - on toiminnoiltaan kattava
  - Helppokäyttöinen, helppoa omaksua jne
- Laadunvarmistuksella on kaksi tehtävää:
  1. ehkäistä virheiden synty
  2. löytää syntyneet virheet mahdollisimman aikaisin

Laadunvarmistus (*quality assurance*)

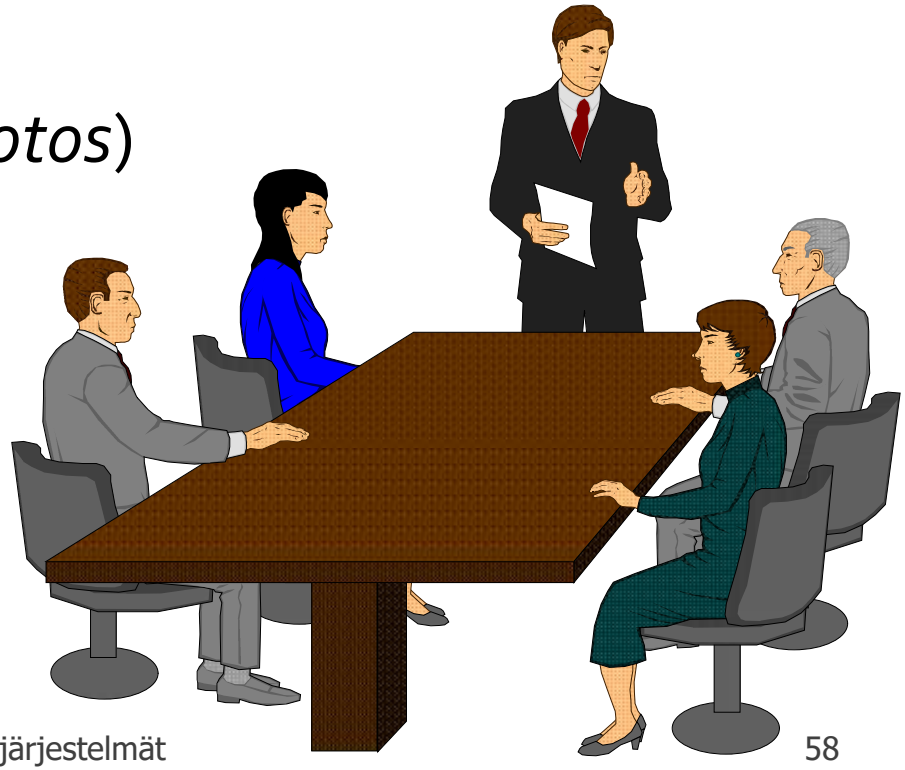
- laatujärjestelmän yksi olennaisimmista osista
- ***tuotteen laatu***
  - vaihetuotteiden laadunvarmistus
    - ohjelmistotuotannossa kaikki tuotetut dokumentit
  - lopputuotteiden laadunvarmistus
    - Ohjelmistotuotannon lopputuotteen ”järjestelmä” erilaisuus
- ***toiminnan laatu***
  - prosessit, menetelmät
- ***työn laatu***
  - henkilöistä riippuvaa
- **johtamisen laatu**

Tuotteen laadunvarmistus

- tavoite: virheiden estäminen
  - *tarkastus* (*inspection*)
    - soveltuu kaikille dokumenteille
    - paras tapa vähentää lopputuotteen vikoja!
- tavoite: virheiden etsiminen
  - *testaus*
    - soveltuu vain ohjelmille
    - kuuluu usein elinkaaren toimintoihin
    - testauksen suorittaminen helppoa, mutta suunnittelu vaikeaa etenkin reaaliaikajärjestelmissä

Tuotteen laadunvarmistus

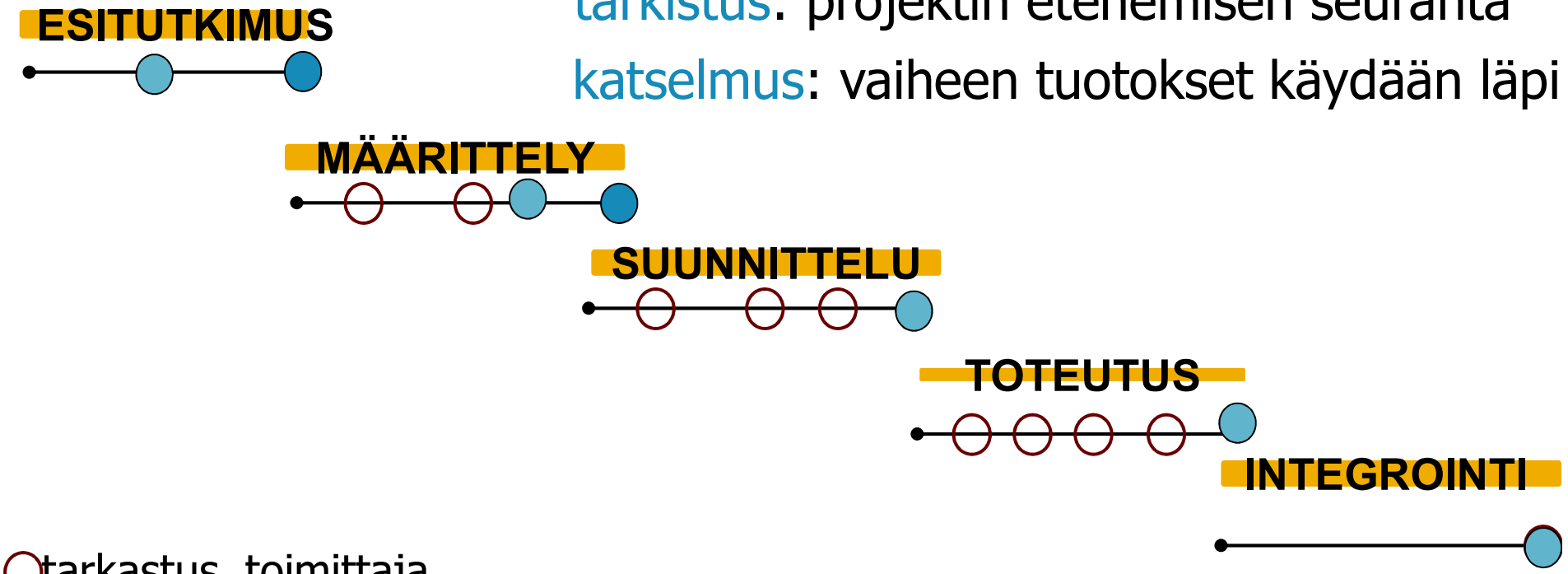
- projektin seuranta
  - tekniset katselmukset (*technical review*)
  - vaiheen tuote (*vaihetuotos*) käydään läpi ja hyväksytään
  - todetaan vaihe päättyneeksi
    - Maksupostit?



Projektin katselmukset ja tarkistukset

tarkistus: projektin etenemisen seuranta

katselmus: vaiheen tuotokset käydään läpi



tarkastus, toimittaja

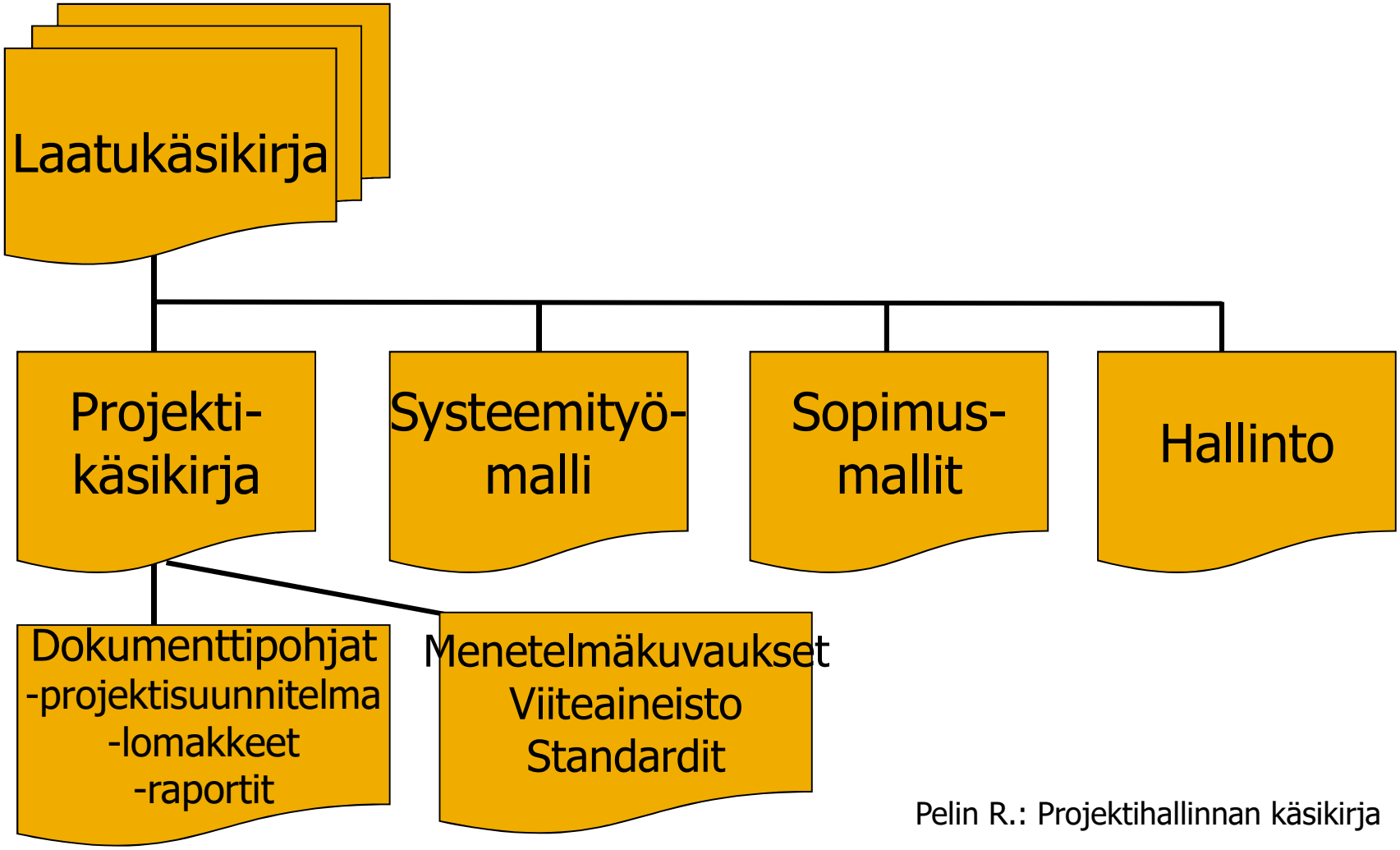
katselmus, toimittaja

katselmus, asiakas mukana

V&V - verification & validation

- *verifiointi* l. *todentaminen*
  - "Are we building the product right?"
- *validointi* l. *kelpoistaminen*
  - "Are we building the right product?"

Tietojärjestelmäprojekteja toimittavan yrityksen laatuohjeisto



Pelin R.: Projektihallinnan käsikirja

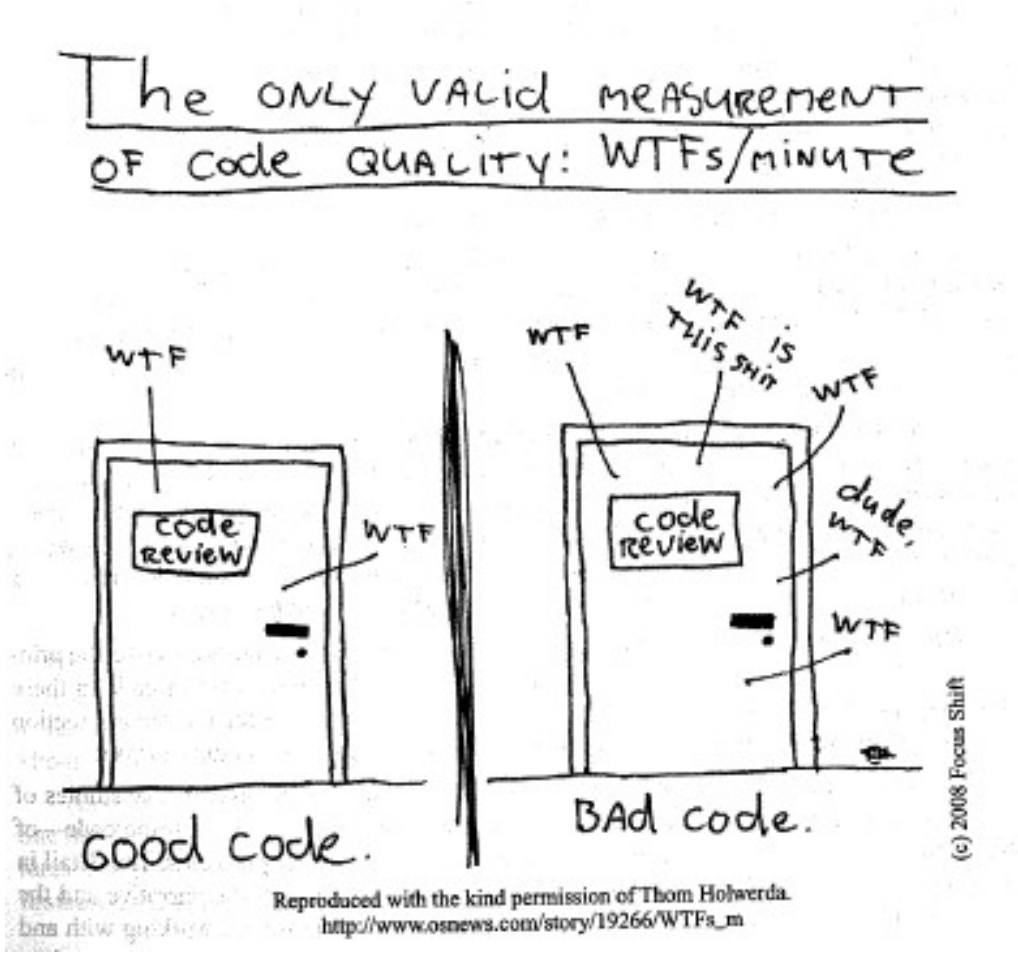
Mistä liikkeelle?

- Toimintotapojen vakiointi
  - omat
  - teamin
  - yrityksen
- Suunnittele mitä teet, tee mitä suunnittelet
- Jos lähdet nollasta, lähde liikkeelle pienestä:
  - ohjelmointi:
    - yhtenäinen muuttujien nimeäminen (koodi on tehty luettavaksi)
    - dokumentoi arkkitehtuuri
  - projekti:
    - tee muistiot kaikista kokouksista
    - tee lista asiakasvaatimuksista
    - kirjaa kaikki muutospyynnöt

11.5 Mittaaminen

The key of controlling anything is measurement.

The only "valid" measurement of code quality: WTFs/minute



Laadun mittaaminen

- ”Mitä ei voi mitata, sitä ei voi ohjata”
- Laatu edellyttää mittaamista, ilman mittaamista ja mittareita et voi luotettavasti ja todistettavasti parantaa laatua.
- Mittarit tuottavat oikein käytettynä faktoja.
- Ilman mittareita mielipiteet, selittelyt ja politikointi pääsevät temmeltämään, päätöksenteosta tulee perusteetonta ja organisaatio menettää elinvoimansa

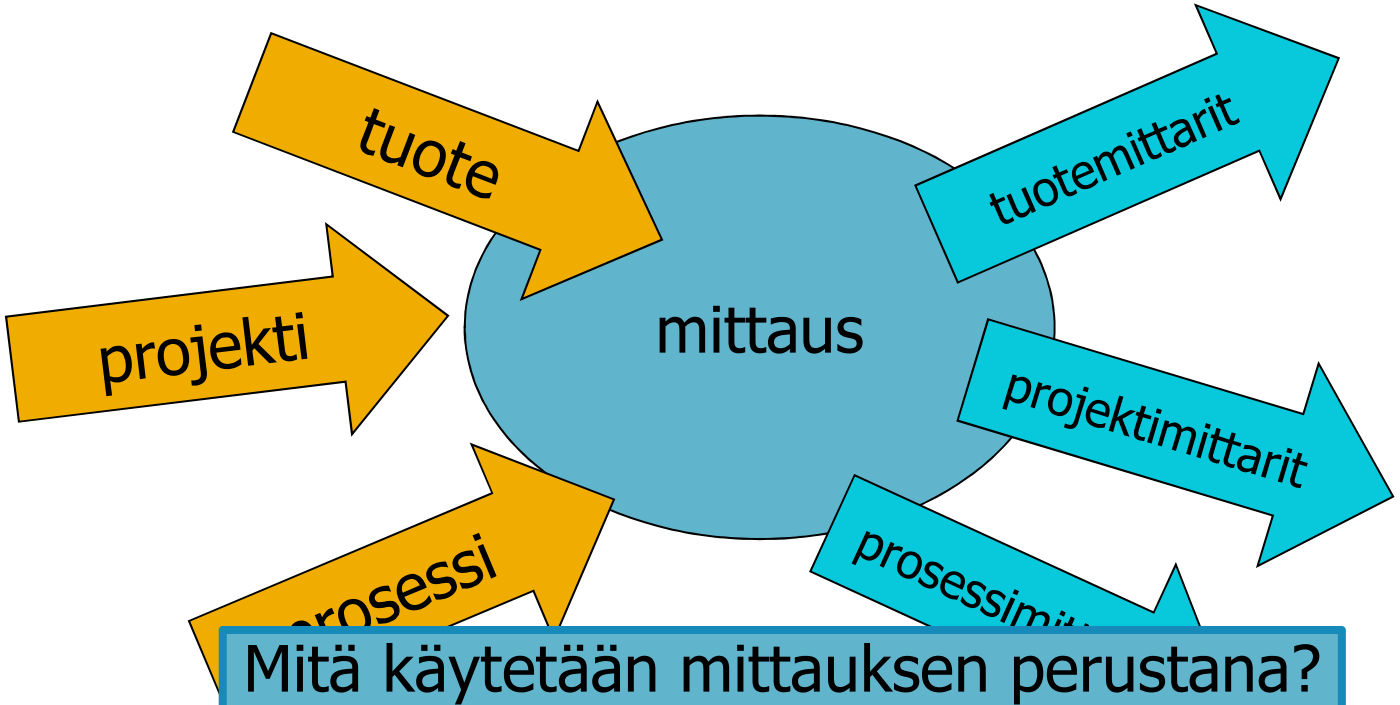
Mittareiden tarkoitus

- Mittareiden tarkoitus on:
  - Nykypäivän ymmärtäminen ja tulevaisuuteen tähtääminen
    - Edistyminen, tavoitteet ja vertailut
    - Ennakointi
  - Päätöksenteon ja toiminnanohjauksen tuki
    - Objektiivista numerotietoa
  - Tärkeää organisaation parantamisen kannalta
    - Tapahtumien ymmärtäminen
    - Muutosten vaikutusten arviointi
    - Ongelmien varhainen paljastaminen
    - Kehityskohteiden tunnistaminen
    - Prosessin kehittäminen

Miksi mitata?

- mittaus palvelee laadunvarmistusta, toiminnan ohjausta ja jatkuvaa parantamista
- mittausten avulla voidaan arvioida toiminnan tehokkuutta
- mittaus luo pohjaa tehokkaalle viestinnälle ja strategioiden käytäntöön viemiselle
- mittaus muuntaa strategian mitattaviksi tavoitteiksi ja tunnusluvuiksi
- mittaus mahdollistaa prosessien ja kannusteiden kytkennän

Mitä mitata ohjelmistotuotannossa?



Mitä mitata ohjelmistoissa

- Ohjelmistoista voidaan mitata monia asioita, tyypillisiä ehdokkaita:
  - Rivien määrä (lines of code)
  - Koodipolut (code paths)
  - (defect rates)
  - Muutosten määrä (change rates)
  - Projektiin käytetty aika (project elapsed time)
  - Budjetti (budget expended)

Hyvä mittari

- hyväksytään ja sillä on merkitystä
- kertoo, kuinka hyvin päämäärät ja tavoitteet saavutetaan
- osoittaa joko trendin tai tason
- on yksinkertainen, ymmärrettävä, looginen ja toistettavissa
- on ristiriidattomasti määritelty
- on kustannustehokas tietojen keruun kannalta
- on oikea-aikainen
- on herkkä

Mittaamisen ongelmia

- Kvalitatiiviset vs kvantitatiiviset
- Aiheuttaa kustannuksia
- Käytännössä kovin luotettavien mittareiden löytäminen
- Asenneongelmat, ihmiset vastustavat mittaamista:
  - ”Epätarkka”, ”hukkatyötä”, ”voi käyttää/tulkita väärin”, ”mahdotonta”
- Ihmiset sopeuttavat toimintonsa mittareihin
  - Vertaa oppimiseen: se opitaan mitä kokeessa kysytään
  - Tuottavuutta mitataan koodirivien määrällä → fiksua vai ei
  - Virheiden löytyminen → koodari koettaa ”piilottaa” virheitään

Prosessin mittareiden tasot

Mittausjärjestelmän olisi tuettava prosessiajattelua!

Prosessien suorituskykyä pitäisi mitata myös toiminnan aikana eikä vain "outputeja" valvomalla!

Mittausjärjestelmän tulisi olla hierarkinen kuten on itse prosessien rakennekin!

Yritystason mittaristo
- strategisesti tärkeät asiat

Prosessitason mittaristo

Toimintotason eli osastotason mittaristo

Mittauspyramidin rakentaminen

Ref. Lynch & Cross, 1992

Ryhmätehtävä vk 40/2009

- Ohjelmistotuotannossa laadun mittaus voi kohdistua: tuotteeseen, projektiin tai prosessiin.
  1. Cooldog: Mitä mittareita ohjelmistotuotteen laadun mittaamisen käytetään? Miten tuloksia hyödynnetään?
  2. Opossumi: Mitä mittareita ohjelmistoprojektin laadun mittaamisen käytetään? Miten tuloksia hyödynnetään?
  3. CC: Mitä mittareita prosessin laadun mittaamisen käytetään? Miten tuloksia hyödynnetään?
  4. Adagroup: Voivatko mittarit&mittaaminen jotenkin auttaa vastaamaan kysymykseen "Milloin ohjelmisto on tarpeeksi hyvä?"
- Kaikki vastaavat: Miksi mitataan eli mitä mittaamisella tavoitellaan?

Ryhmätyö vk 40/2009

- Miettikää ja dokumentoikaa ryhmissä seuraavia asioita (mitä mitataan, miten, milloin, miksi):
  - Cooldog: Ohjelmistotuotteen laadun mittaaminen
  - Opossumi: Ohjelmistoprojektin laadun mittaaminen
  - CC: Ohjelmistotuotantoprosessin laadun mittaaminen
  - Adagroup: Mitä erilaisia laadun mittareita käytetään ohjelmistofirmoissa & voivatko mittarit&mittaaminen jotenkin auttaa vastaamaan kysymykseen "Milloin ohjelmisto on tarpeeksi hyvä"

Esimerkkejä prosessin mittareista

- projektien aika- ja kustannusarvioiden toteutuminen (ero suunniteltuun)
- eri vaiheissa löytyneet virheet
- katselmoinneissa löytyneet virheet
- eri vaiheissa syntyneet virheet
- työn tuottavuus (SLOC/pv)
- laskutuskelvottomat tunnit
- takuutyön osuus (kustannus)

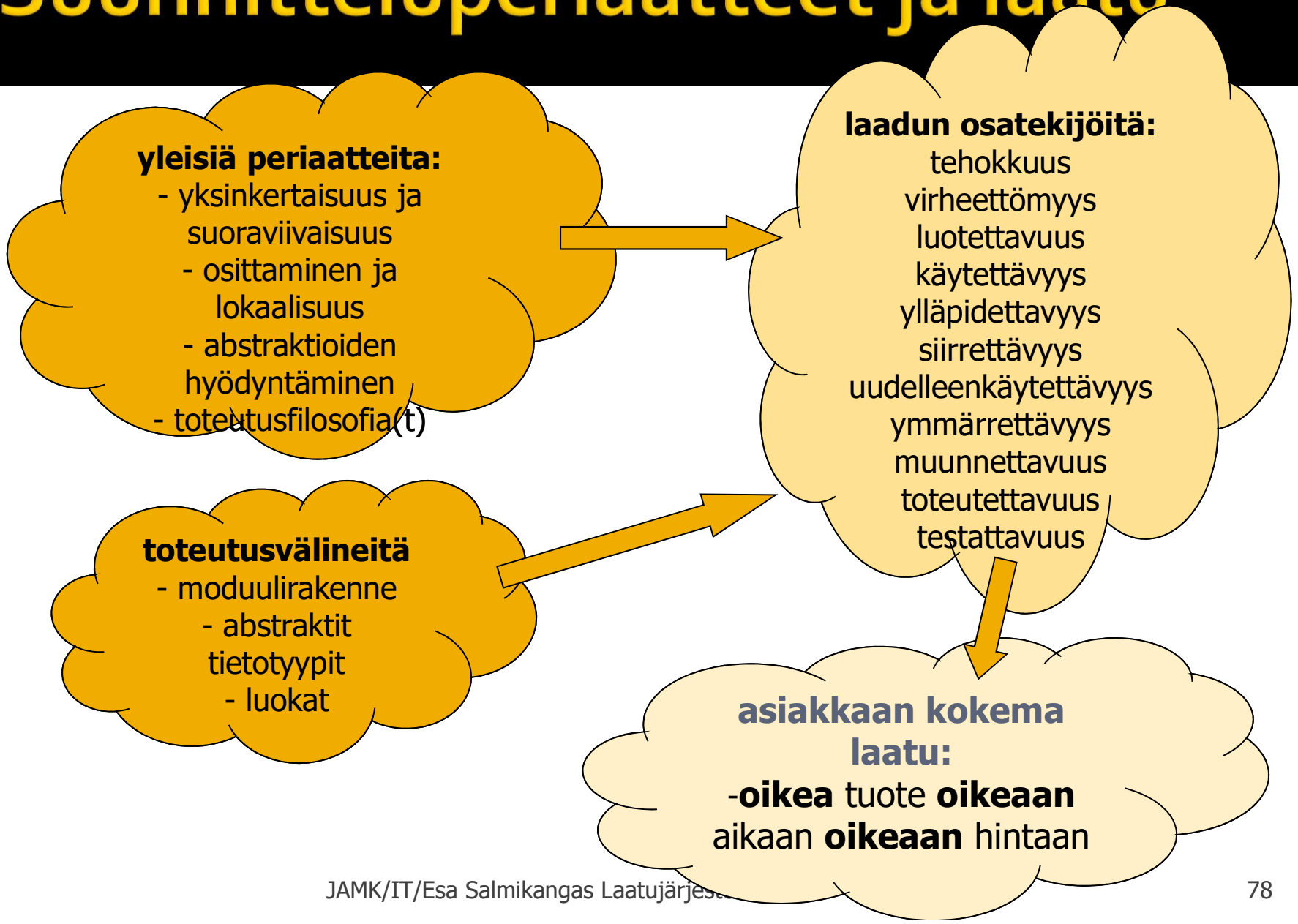
Esimerkkejä tuotteen mittareista

- ohjelmiston koko
- virheiden määrä ohjelmarivejä kohden
- mutkikkuusmittarit (ks. luku 15)
- palveluaste (% kokonaisajasta)
- vikaantumisväli (MTBF)
- korjausaika (MTTR)
- Tehokkuusmitat
- asiakastyytyväisyys(kysely)

11.x Mitä on ohjelmistojen laatu?

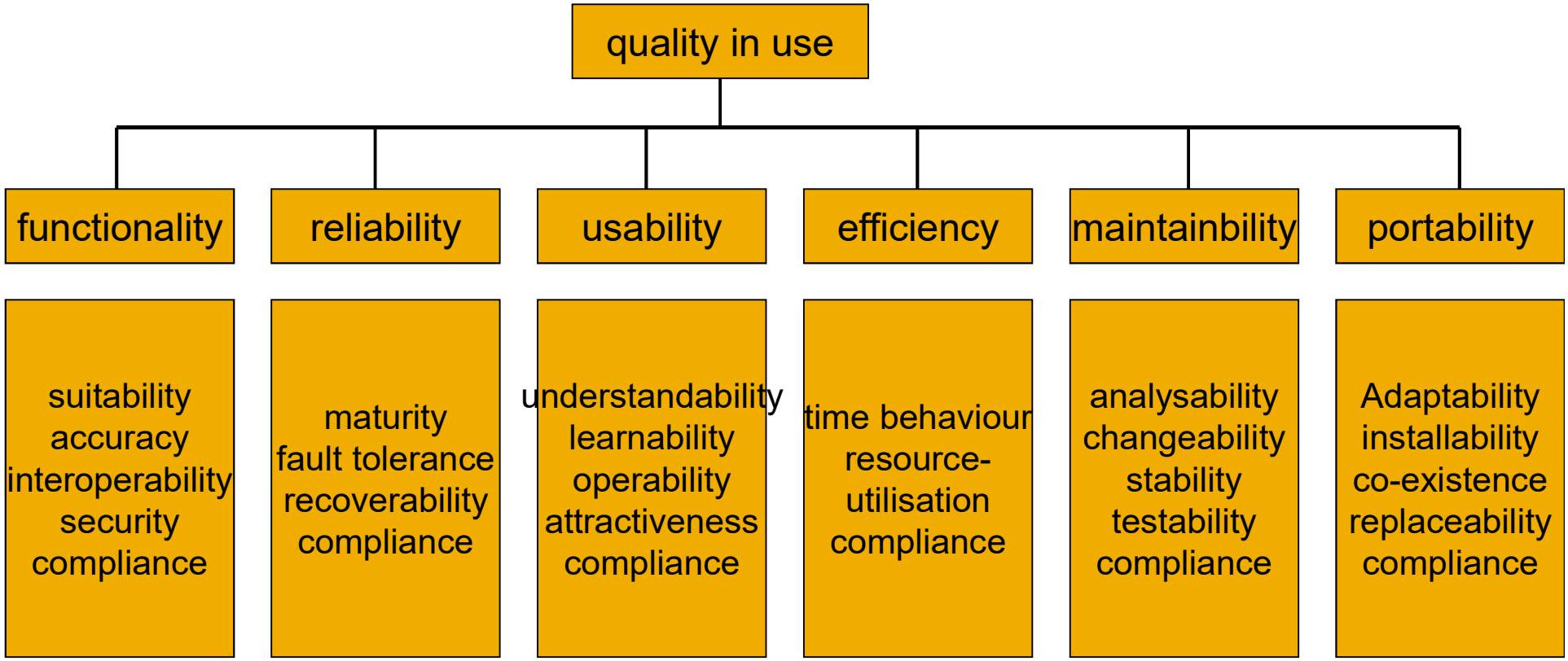
Mistä koostuu ohjelmistojen laatu?

Suunnitteluperiaatteet ja laatu

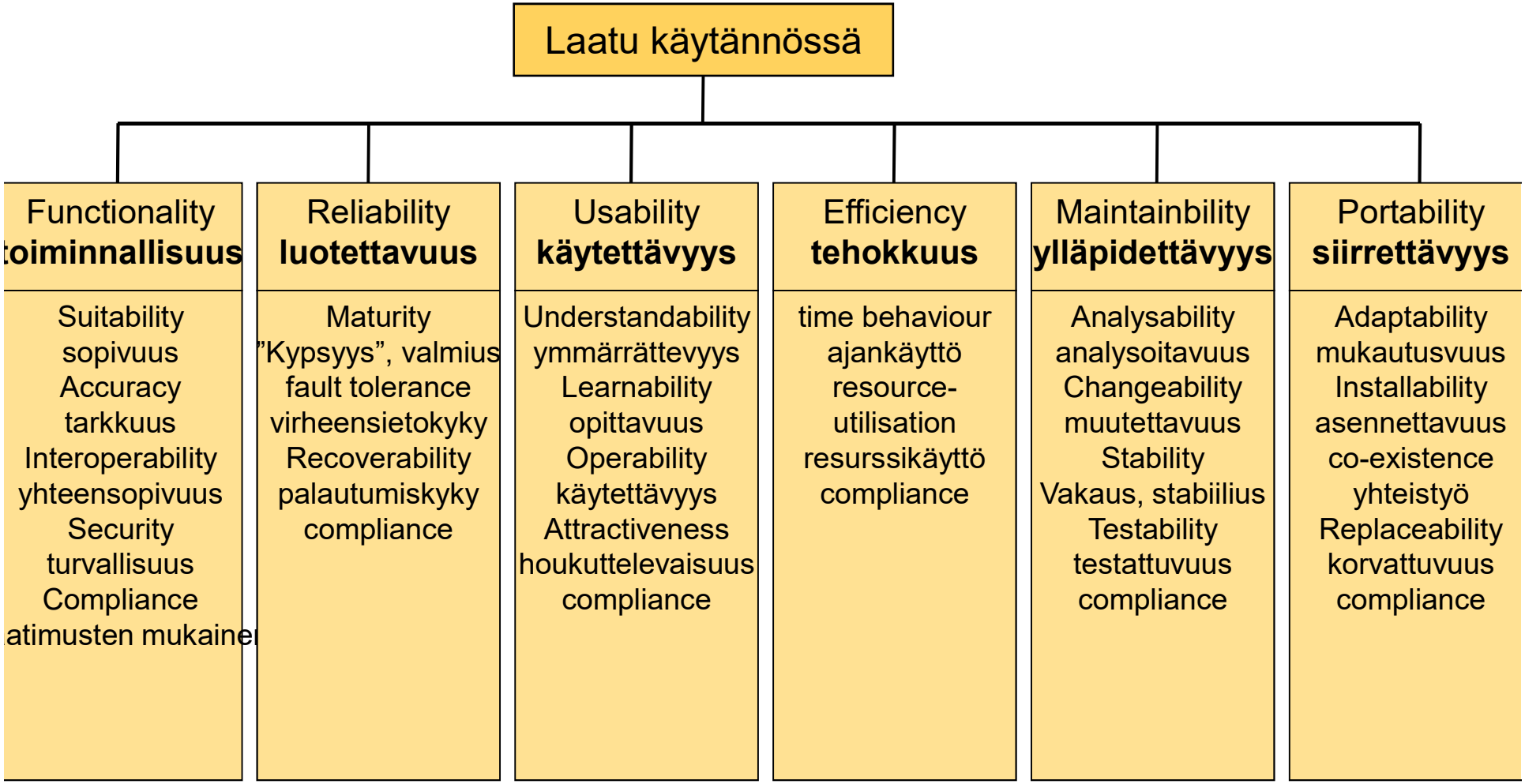


Ohjelmistojen laadun "tekijät"
Software quality characteristics

Software quality characteristics (ISO/IEC 9126 Quality Model)



Mistä ohjelmistolaatu koostuu?



lähde/source: User interfaces for all

Software quality attributes

source Ian Sommerville: Software Engineering

Safety	Understandability	Portability
Security	Testability	Usability
Reliability	Adaptability	Reusability
Resilience	Modularity	Efficiency
Robustness	Complexity	Learnability

resilience [rɪ'zɪliəns[i]] *s* (*myös* resiliency) ***1*** (materiaalin) joustavuus, kimmoisuus ***2 kuv*** sitkeys, sinnikkyys, lannistumattomuus *a woman of great ~* todella sinnikäs nainen ***3*** (*~ to*) vastustuskyky[isyys], sietokyky, kestävyys *his ~ to stress* hänen stressinsietokykyänsä

11.x2 Mitä on ohjelmistotuotannon laatu?

eli mistä koostuu "ohjelmistojen tekemisen" laatu

Ohjelmistotuotannon vaiheet ja laatu

- laadun perusprinsiippi:
  - "nykyaikaisessa laatujohtamisessa lähdetään siitä että laadukas prosessi tuottaa laadukkaita lopputuotteita"
  - → niin tärkeää on pyrittäessä hyvälaatuiseen lopputulokseen:
    - Tunnistaa mistä vaiheista (osista) ohjelmistotuotanto koostuu.
    - Tunnistaa miten vaihe vaikuttaa lopputuotteeseen ja seuraaviin vaiheisiin.
    - Analysoida tunnistettujen vaiheitten hyvyyttä/heikkoutta.
    - Valita ne vaiheet joihin a) voi vaikuttaa b) kannattaa vaikuttaa.
    - Vaikuttaa eri laatua parantavilla toimenpiteillä valittuihin vaiheiseen.
    - Mitata toimenpiteiden vaikutusta ja reagoida.

Kahdeksan ohjelmistotuotannon peruselementtiä (Wasserman 1996) #1

1) Abstraktio
   - Tarvitaan tapoja tarkastella sekä ongelmaa että ratkaisuja useilla tarkkuuden tasoilla.
2) Kuvauskielet ja -menetelmät
   - Tarvitaan tapoja kommunikoida ja mallintaa sekä ongelmaa että ratkaisua.
3) Ohjelmistoarkkitehtuuri
   - Tarvitaan tapoja ongelman sekä ratkaisun osittamiseen ja näin syntyvän rakenteen kuvaamiseen.
4) Ohjelmistoprosessi
   - Tarvitaan toistettava, mitattava ja ennustettava tapa toimia.
5) Uudelleenkäyttö
   - Tarvitaan mekanismeja ja käytäntöjä, joilla voidaan aiemmin tehtyä työtä hyödyntää myöhemmin, kaikissa työvaiheissa ja kaikilla tasoilla.
6) Mittaaminen
   - Tarvitaan tapoja mitata sekä tuotetta että toimintaa, jotta laadun, toiminnan ohjauksen ja ennustettavuuden parantaminen olisi mahdollista.
7) Työkalut
   - Kaikkia toiminnan vaiheita tukemaan tarvitaan hyvät työkalut ja niiden tehokas käyttö (esim. IDE,versionhallinta, testausympäristöt).
8) Prototyypitys
   - Rakennettavaa järjestelmää ja sen osia on päästävä kokeilemaan, jotta ymmärretään, mitä oikeastaan halutaan. On realistista olettaa, ettei ilman ratkaisun prototyyppiä tiedetä, millainen ratkaisu ongelmaan tarvitaan.

How do software engineers spend their time on the job? #1

- Ohjelmistotuotannossa, koodaus on vain osa ohjelmistotuotteen tekemisestä…
- → Laplanten mukaan ”Ohjelmistoinsinöörit todennäköisesti käyttävät vähemmän kuin 10% työajastaan koodin kirjoittamiseen. Loput 90% he käyttävät muihin tärkeämpiin aktiviteetteihin!”

Lähde: P.A.Laplante, What every should know about Software Engineering

Activities of software engineers First seven: one to seven

1. Eliciting requirements
2. Analyzing requirements
3. Writing software requirements document
4. Building and analyzing prototype
5. Developing software designs
6. Writing software design documents
7. Researching software engineering techniques or obtaining information about application domain

1. Vaatimusten selvittäminen
2. Vaatimusten analysointi
3. Vaatimusmäärittely dokumenttien kirjoittaminen
4. ...

Lähde: P.A.Laplante, What every should know about Software Engineering

Activities of software engineers
Second seven: eight to fourteen

8. Developing test strategies and test cases.
9. Testing the software and recording the results.
10. Isolating the problem and solving it.
11. Learning to use or installing and configuring new software and hardware tools.
12. Writing documentation such as users manual.
13. Attending meetings with colluages, customers, supervisors etc
14. Archiving software and readying it for distribution.

8. Testausstrategioitten ja testitapausten kehittäminen

Lähde: P.A.Laplante, What every should know about Software Engineering

11.6 Laatujärjestelmän auditointi

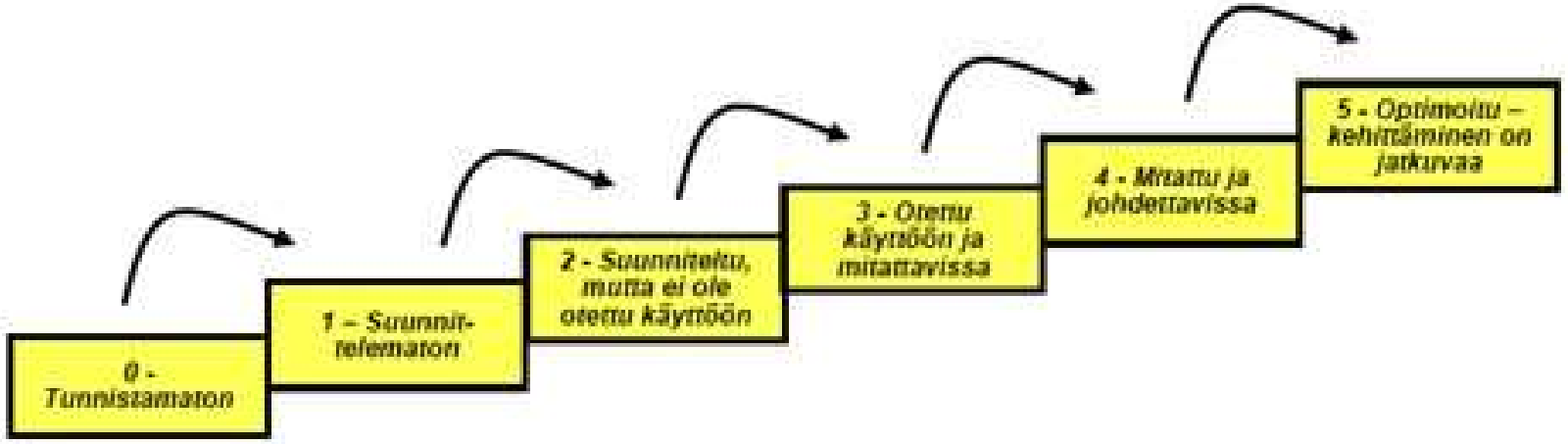
Auditointi

- Puhutaan ns “auditoitu laatujärjestelmä”
- Ulkopuolisen/riippumattoman osapuolen tekemä laatujärjestelmän arviointi
  - De Norske Veritas tms
- Verrataan yrityksen toimintaa johonkin vertailukriteereihin…

11.7 Laatujärjestelmän kehittäminen

Laatujärjestelmän kehittäminen

- Toimintaa jossa yrityksen toimintaa kehitetään tavalla joka vaikuttaa positiivisesti yrityksen toimintaan ja sitä kautta lopputuotteisiin/palveluihin
- CMM = Capability Maturity Model



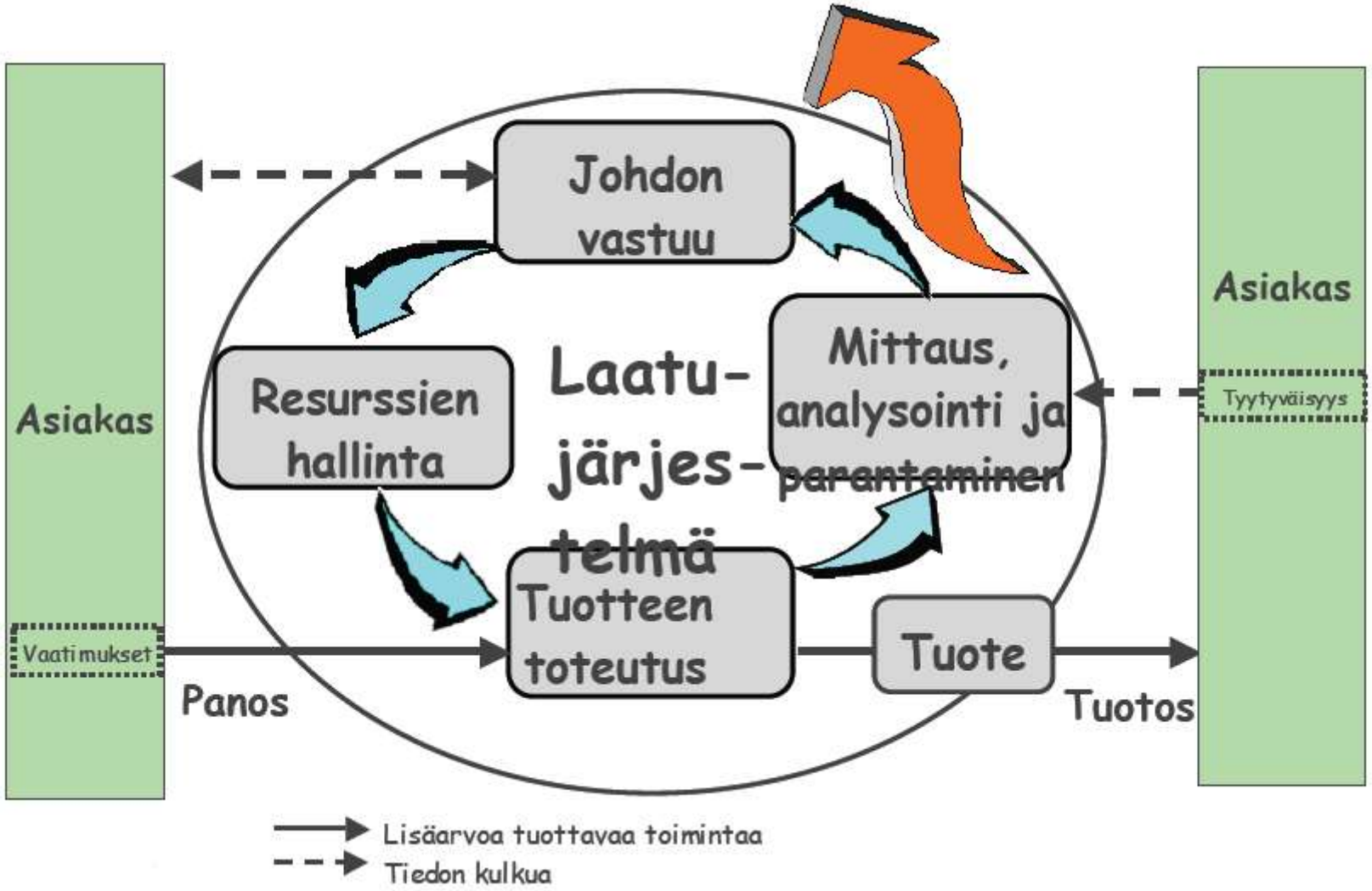
Merkittävin laatujärjestelmiin vaikuttava standardisarja

11.8 ISO 9000 -standardisarja

ISO 9000

- Miksi?
  - Alkuperäinen tarkoitus oli ”nostaa paikallisen teollisuuden kilpailukykyä”
- Mikä
  - Geneerinen, kaikille teollisuusaloille, käytetään myös ohjelmistoalalla
- Mitä
  - Sisältää kolme varsinaista standardia

ISO9001 rakenne



ISO20000

- **ISO/IEC 20000** is the first international standard for IT Service Management.
- lisää: http://en.wikipedia.org/wiki/ISO/IEC_20000

11.9 Muita laatujärjestelmän arviointi ja kehittämismalleja

TQM Total Quality Management

- TQM
  - a management strategy aimed at embedding awareness of quality in all organizational processes.
  - widely used in manufacturing, education, government, and service industries
- Ei ole vakiintunut ohjelmistopuolelle, lähin vastaavaa CMM Capability Maturity Model

source: http://en.wikipedia.org/wiki/TQM

CMM = Capability Maturity Model

- also known as the Software CMM (SW-CMM)
- first described by Watts Humphrey in his book Managing the Software Process [1].
- The CMM is a process model based on software best-practices effective in large-scale, multi-person projects.
- The CMM **has been retired** and not been updated in over 10 years.
  - → CMM has been superseded by CMMI (Capability Maturity Model Integration).

CMMI

source: http://www.sei.cmu.edu/cmmi/general/

- Capability Maturity Model Integration
  - Widely used, introduced 2002
  - CMMI is a process improvement approach that provides organizations with the essential elements of effective processes. It can be used to guide process improvement across a project, a division, or an entire organization.
  - more: CMMI-overview06.pdf

CMMI Suomes

- On käytössä
  - ” Software Engineering Institu
    prosessit ja vahvistanut yhtiö
    (Capability Maturity Model In
  - Esimerkiksi Intiassa CMMI-ma
    toimittaja.

ITViikko 22.4.2008

IT VIIKKO Etusivu

Etusivu | Uutiset | Kolumnit | Seminaarit | Keskustelut

Capgemini pääsi kypsyystasolle kolme

Olli Häkämies/Lehtikuva

Markus Sell, Capgemini Finlandin toimitusjohtaja.

Software Engineering Institute (SEI) on arvioinut Capgemini Finlandin prosessit ja vahvistanut yhtiön saavuttaneen CMMI-prosessikehitysmallin (Capability Maturity Model Integration) kypsyystason 3.

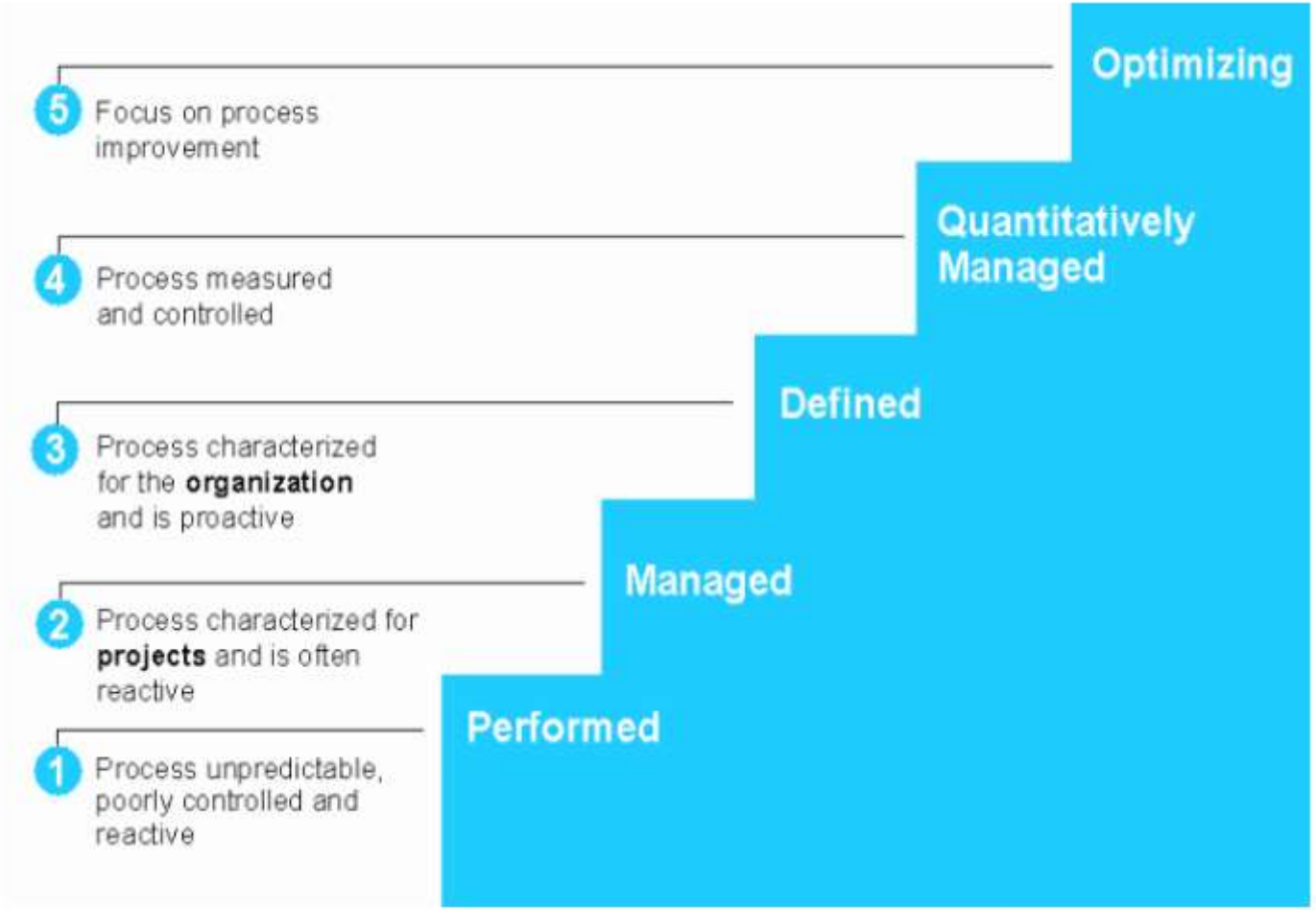
Jarmo Lahti
22.4.2008 13:05
0

0
Kommentoi
Lähetä
Tulosta (HTML)
Tallenna (PDF)
Del.icio.us
Facebook

CMMI-malli varmistaa yhtenäisen ja laadukkaan prosessin tietojärjestelmien koko elinkaaren hallintaan. Se sisältää strukturoidut mallit ja parhaat käytännöt tietojärjestelmien toimittamista, kehitystä ja hallintaa varten.

- Tietojärjestelmän laatuun vaikuttaa keskeisesti sen toimittamisessa, kehittämisessä ja ylläpidossa käytetyn prosessin laadukkuus. Tietojärjestelmäprojektissa uusin

CMMI mallin tasot



SPICE

- SPICE - Software Process Improvement and Capability dEtermination
- Hanke ISO:n piirissä yhteisen standardin kehittämiseksi ohjelmistotuotannon prosessien arviointiin ja parantamiseen (ISO15504/SPICE98)
  - ISO/IEC 15504 pohjautuu ISO 12207:sta ja käyttää monia CMMIn ideoita.
- Hankkeen tavoitteena on:
  - tarjota nopeasti ratkaisu todelliseen arviointitarpeeseen
  - yhdistää maailman eri puolilla toimivien asiantuntijoiden osaaminen ja nykymalleista saadut kokemukset
  - kokeilla saavutettuja tuloksia käytännössä
  - antaa ISO:n jäsenvaltioille mahdollisuus vaikuttaa asioihin
  - tehdä uusi standardi tunnetuksi käyttäjille

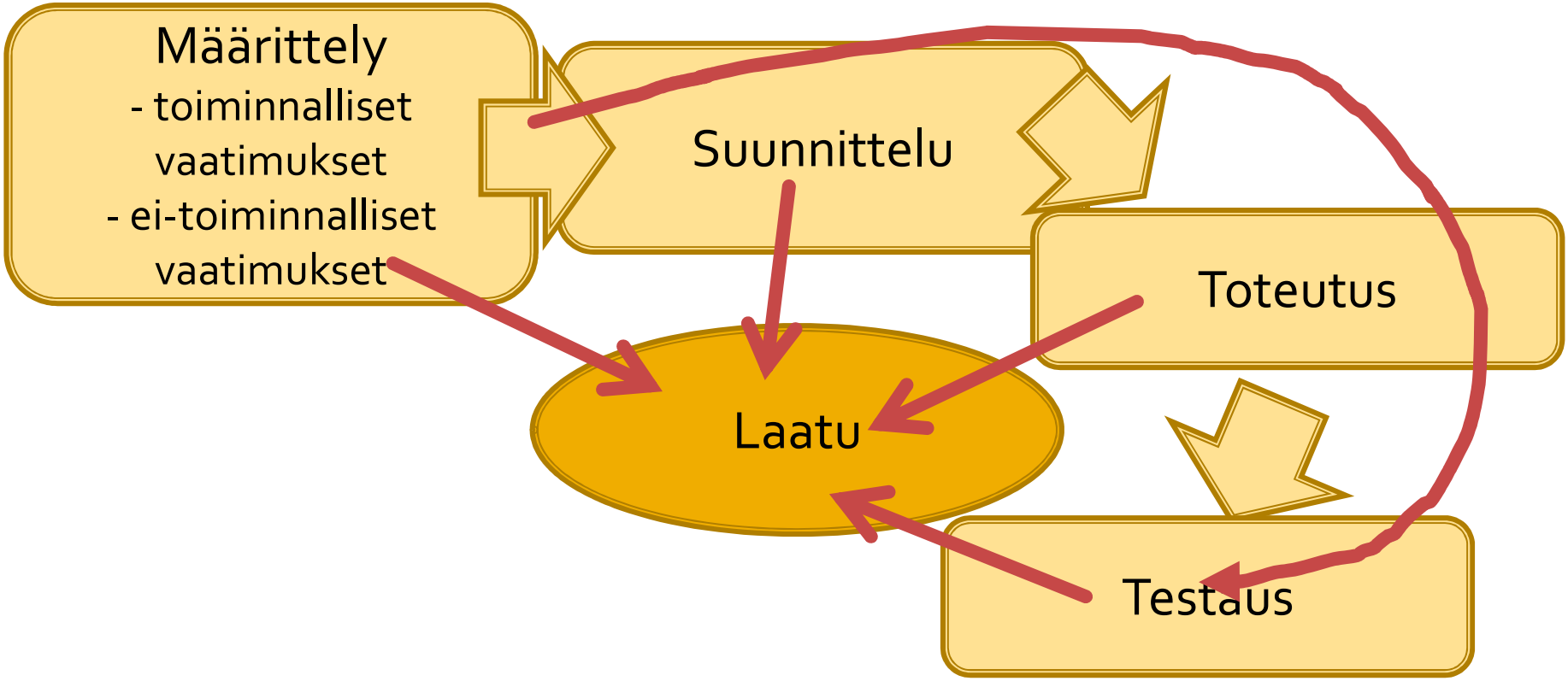
11.xx Miten laatua "tehdään" ohjelmiin

"The cat ate my source code."

One of cornerstones of the pragmatic philosophy is the idea of **taking responsibility** for yourself and your actions in terms of your career advancement, your project, and your day-to-day work.
A.Hunt, D.Thomas: "Pragmatic Programmer"

Laatu ”tehdään” koko tuotantoprosessin aikana

- Sitä ei vaan ”liimata” koodausvaiheessa…



The Joel Test: 12 Steps to Better Code

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

http://www.joelonsoftware.com/articles/fog0000000043.html

Ohjelmoinnin "laatupyrkimykset"
(by esa tietotekniikan esihistorialliselta ajalta ennen Windowsia/Linuxia)

- Pyrkimys modulaarisuuteen
- Pyrkimys luettavuuteen ja yleisiin ratkaisuihin
- Pyrkimys järkevään kommentointiin
- Pyrkimys ylläpidettävyyteen

Ohjelmoinnin "Laatupyrkimykset" 1

- Pyrkimys modulaarisuuteen
  - yhteen asiaan liittyvät koodit on yhdessä paikassa
  - isot ohjelmat koostuvat joko aliohjelmakutsuista tai "palikoista" jota on helppo napsia käyttöön ja pois käytöstä
  - jos käytät samaa koodin pätkää enemmän kuin kerran tee siitä aliohjelma ja kommentoi sen käyttö
- Pyrkimys luettavuuteen ja yleisiin ratkaisuihin
  - vaikka vähän enemmän koodirivejä, jotta koodi olisi mahdollisimman helppo lukea myöhemmin
  - ei mitään ihmeellisiä koodin lyhentämiskikkailuja luettavuuden kustannuksella
  - jatkukoon koodi mieluummin alaspäin kuin oikealle

Ohjelmoinnin ”Laatupyrkimykset ” 2

- Pyrkimys järkevään kommentointiin
  - käytä kuvaavia ohjelmien, funktioiden ja muuttujien nimiä, vaikkakin olisivat pitkiä, kunhan eivät ole ylipitkiä
  - dblBruttoPalkka on parempi kuin bp
- Pyrkimys ylläpidettävyyteen
  - tee koodia jota voi muuttaa ja korjata
    - selkeä (ja dokumentoitu) arkkitehtuuri, modulaarinen rakenne
    - ”vakiot pysyy vakiona ja muuttujat muuttuu”
  - joudut kuitenkin ylläpitämään omaa tai jonkun toisen tekemää järjestelmää ennemmin tai myöhemmin

Ohjelmoinnin yleiset laatuvaatimukset #1

lähde: ote ohjelmointiyrityksen "XYZ" laatukäsikirjasta vuodelta 1990

- 1. TARKOITUS
  - Yleisiä ohjelmoinnin laatuvaatimuksia on tarkoitus käyttää määrittelyvaiheessa tehtävän ohjelmointityön prioriteettejä asetettaessa sekä valmistuneen ohjelmiston lopullista laatua arvioitaessa.
- 2. LAAJUUS
  - Ohjelmoinnin yleiset laatuvaatimukset koskevat kaikkea ohjelmointi- ja järjestelmäkehitystyötä, jota tehdään.
- 3. VASTUU
  - Ohjelmoitsija vastaa siitä, että yksittäinen ohjelma täyttää asetetut vaatimukset ja projektipäällikkö vastaa siitä, että kokonaisuus täyttää asetetut vaatimukset.

Ohjelmoinnin yleiset laatuvaatimukset #2

4. MENETELMÄ

- Yleisesti ottaen ohjelman laadulla tarkoitetaan sitä, kuinka hyvin ohjelma vastaa sille esitettyjä vaatimuksia ja kuinka korkealle vaatimukset on asetettu. Vaatimustaso tulisi asettaa ennen ohjelmointityön aloittamista ja kirjata se osaksi projektisuunnitelmaa.
- Vaatimustekijät ja niiden keskinäinen painotus voivat vaihdella tapauskohtaisesti. Myös painotukset tulisi kirjata projektisuunnitelmaan.
- Laatuvaatimukset on jaettu kahteen kohteesta riippuvaan ryhmään*:
  - A) ohjelman 'käyttäjät'
  - B) ohjelman 'tekijät'
    - *Kuitenkaan kaikista laatuvaatimuksista ei voi sanoa, että ne kuuluisivat yksikäsitteisesti jompaan kumpaan ryhmään vaan ne saattavat olla molemmilla ryhmille tärkeitä.

Ohjelmoinnin yleiset laatuvaatimukset #3-1/4

1) Oikeellisuus (*correctness*) (Ryhmä A,B)
   - ehdottomasti aina tärkein, sillä virheellisesti toimivalla ohjelmalla ei ole käyttöä
2) Luotettavuus (*reliability*) (Ryhmä A,B)
   - ohjelman arvo laskee, jos käyttäjä ei voi luottaa ohjelman antamiin palautteisiin
3) Käyttökelpoisuus (*validity*) (Ryhmä A)
   - oikeinkaan toimivalla ohjelmalle ei ole käyttöä, ellei se ratkaise käyttäjän ongelmaa
4) Tehokkuus (*efficiency*) (Ryhmä A,B)
   - ohjelman olisi oltava tehokas muistitilan ja suoritusajan suhteen

Ohjelmoinnin yleiset laatuvaatimukset #3-2/4

5) Luettavuus (*readbility*) (Ryhmä B)

- muidenkin kuin ohjelman tekijän on tarvittaessa voitava ymmärtää ohjelman ajatus ja toiminta koodin ja muuhun dokumentaation avulla

6) Testattavuus (*testability*) (Ryhmä B)

- ohjelman testaamisessa ja oikeellisuuden varmistamisessa on usein suurempi työ kuin ohjelman tekemisessä

7) Korjattavuus (*debuggability*) (Ryhmä B)

- ohjelmassa esiintyvät virheet tulee voida helposti paikallistaa ja korjata

8) Ylläpidettävyys (*maintainability*) (Ryhmä B)

- ohjelmaa on tarvittaessa voitava sovittaa ja laajentaa tarpeiden ja olosuhteiden muuttuessa

Ohjelmoinnin yleiset laatuvaatimukset #3-3/4

9) Siirrettävyys (*portability*) (Ryhmä A,B)

- koneet ja laitteet kehittyvät ja halpenevat nopeammin kuin ohjelmat, koska ne ovat yksinkertaisempia --> ohjelmia tulisi voida siirtää koneista ja ympäristöistä toiseen

10) Sovitettavuus (*adaptability*) (Ryhmä B)

- käyttötarkoituksen ja tarpeiden ajallinen ja paikallinen vaihtelu edellyttävät helppoa muunneltavuutta --> ohjelman eri osat toisistaan riippumattomia (mustia laatikoita)

11) Käyttäjäystävällisyys (Ryhmä A)

- tarkoittaa virheiden sietokykyä, visuaalista miellyttävyyttä, toimintojen loogisuutta ja samankaltaisuutta, helppoa omaksuttavuutta

Ohjelmoinnin yleiset laatuvaatimukset #3-4/4

12) Uudelleen käytettävyys (*reusability*)
(Ryhmä B)

- koodin uudelleen käytettävyydellä voidaan seuraavien ohjelmointitöiden ohjelmointiaikaa merkittävästi pienentää

13) Loppukäyttäjän muokkausmahdollisuus
(Ryhmä A)

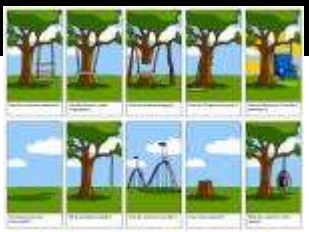
Ohjelmistovirheet

error, mistake, bug,
fault, failure, feature

Ohjelmistovirhe

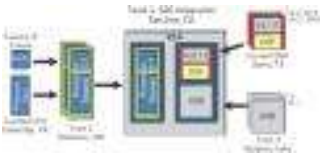
- Virhe (*error*) eli bugi (*mistake, bug*): ihmisen tekemä möhläys, esimerkiksi ohjelmointivirhe tai virhe dokumentissa. Ohjelmointivirheet voidaan jakaa:
  - syntaksi virhe
  - looginen virhe
  - ajonaikainen virhe
- Kun virheellinen ohjelmankohta suoritetaan, se saattaa aiheuttaa vian (*fault*). Järjestelmä on nyt tilassa, joka ei ole sen määrittelyn mukainen.
- Vika puolestaan voi aiheuttaa häiriön (*failure*), joka on vian ilmeneminen järjestelmän ulkoisessa käyttäytymisessä.
- Dokumenttien tulkinnallisuus: asiakkaan mielestä selvä virhe voi olla toimittajan mielestä ominaisuus (*feature*)

Ohjelmistovirheiden aiheuttajia



- vaatimusmäärittelyvirheet
  - virheelliset, puutteelliset, ylimääräiset ominaisuudet
- kommunikointiongelmat
  - asiakas vs kehittäjä; laitetoimittaja vs ohjelmistotoimittaja; toimi alihankkija

- tarkoitukselliset poikkeamat vaatimuksista
  - ohjelmistomoduulien uudelleenkäyttö
  - ajanpuutteesta johtuvat toteuttamattomat osat
  - kehittäjän omat lisäykset



- loogiset suunnitteluvirheet
  - virheelliset algoritmit ja prosessin virheet
  - raja-arvovirheet
  - järjestelmän tilan puuttuvat tarkistukset
  - virhetilanteiden puuttuva käsittely

- koodausvirheet
  - mutta mehän ei niitä tehdä ☺

Ohjelmistovirheiden poisto

Kymmenen kärki - lista

Top 10 (1/4)

- Ohjelmisto-ongelman löytämisen ja korjaamisen kustannukset toimituksen jälkeen ovat usein 100 kertaa kalliimmat kuin sen löytämisen ja korjaamisen kustannukset vaatimus- ja suunnitteluvaiheissa.
  - Epäkriittisissä järjestelmissä kulujen kustannuskerroin 5:1
  - Hyvät arkkitehtuurikäytännöt pienentävät kerrointa jopa suurille, kriittisille järjestelmille
- Nykyiset ohjelmistoprojektit käyttävät noin 40 - 50 % työmäärästä vältettävissä olevaan uusintatyöhön
  - Prosessin kypsyyden, ohjelmistoarkkitehtuurien ja riskianalyysien parantaminen

Boehm,B, V Basili, 2001, Software Reduction Top 10 List: Computer, v. 34

Top 10 (2/4)

- Noin 80 % vältettävissä olevasta uusintatyöstä tulee 20 % virheistä
  - Korjaustyömäärien tallennus seurantajärjestelmään
- Noin 80 % virheistä tulee 20 % moduuleista, ja noin puolet moduleista on virhevapaita
  - Tunnista virheherkkien moduulien tunnusmerkit tietyssä ympäristössä
- Noin 90 % järjestelmän alhaallaoloajasta tulee korkeintaan 10 % virheistä
  - Riskipohjainen testaus – järjestelmän toimintaprofiilit ja korkean riskin skenaarioiden testaus

Boehm,B, V Basili, 2001, Software Reduction Top 10 List: Computer, v. 34

Top 10 (3/4)

- Vertaiskatselmoinnit löytävät 60 % virheistä
  - Vertaiskatselmoinnit, analyysityökalut ja testaus löytävät erityyppisiä virheitä kehityssyklin eri vaiheissa.
- Perspektiivipohjaiset katselmoinnit löytävät 35 % enemmän virheitä kuin kohdistamattomat katselmoinnit
  - Organisaation olemassaoleva virhehistoria
- Kurinalaiset henkilökohtaiset käytännöt voivat vähentää virheiden syntymistaajutta jopa 75 %
  - Cleanroom – prosessi ohjelmistokehitykseen
  - PSP ja TSP

Boehm,B, V Basili, 2001, Software Reduction Top 10 List: Computer, v. 34

Top 10 (4/4)

- Muiden muuttujien ollessa samoja, korkean luotettavuustason ohjelmistotuotteiden kehittäminen maksaa 50% enemmän lähdekoodikomentoa kohden kuin alhaisen luotettavuustason ohjelmistojen kehittäminen.
  - Sen arvoista jos projektiin liittyy merkittäviä toiminta- ja ylläpitokustannuksia
- Noin 40 - 50 % käyttäjäohjelmistoista sisältää oleellisia virheitä

Boehm,B, V Basili, 2001, Software Reduction Top 10 List: Computer, v. 34

Tässä vain lyhyesti esiteltynä,
enemmän Ohjelmistoprojekti-materiaalissa

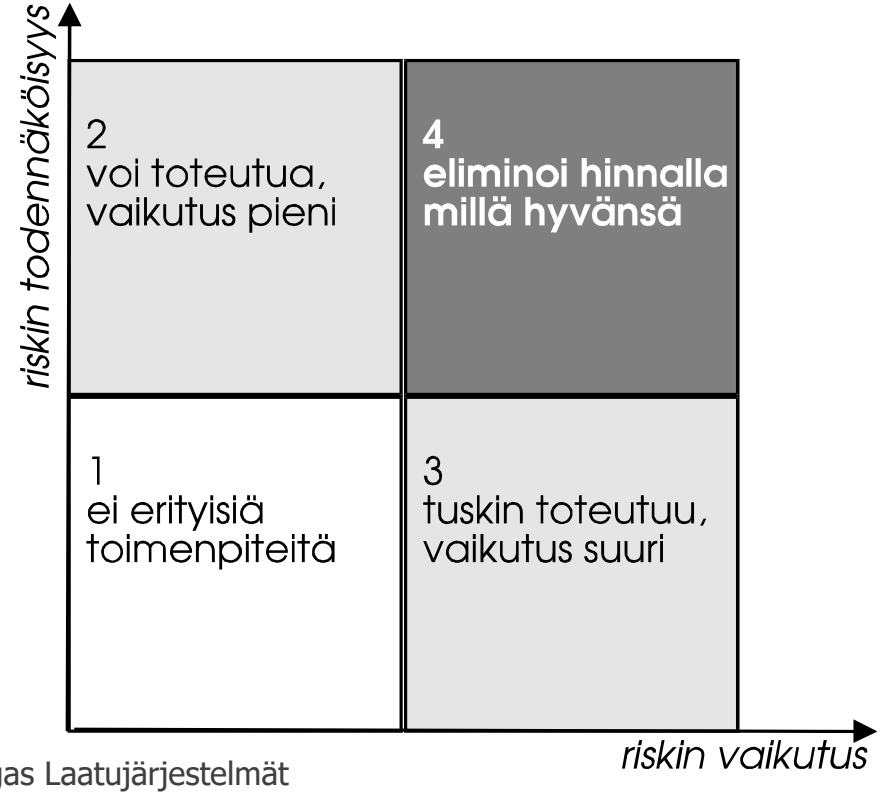
Projektin laatu

Projektin riskien hallinta

- Projektin hallinta on epävarmuuden ja vaihtelevien olosuhteiden hallintaa. Riskien hallinnan tehtävänä on vähentää tätä epävarmuutta.
- Riskien hallinta projektissa on laajempi käsite, johon kuuluu:
  1. riskien analysointi
  2. riskilistan laatiminen
  3. toimenpiteistä sopiminen
  4. seuranta

Riskitaulukko

- Riskien hallinnassa on syytä suunnata energia riskien eliminoimiseen, eikä niiden analysointiin



Riskien analysointi
Sba-menetelmä

- Tietojärjestelmäprojektit ovat usein suuria investointeja. Niiden riskit halutaan tunnistaa ja arvioida etukäteen. Tätä kutsutaan riskien analysoinniksi.

1) Projektin koko:
   Onko järkevä ja kohtuullinen?
2) Atk-kypsyys
   Millaiset ovat valmiudet uuden tietojärjestelmän käyttöön?
3) Tekniikka
   Miten hyvin tunnetaan tekniikka,
   jota tarvitaan uudessa järjestelmässä ?
4) Projektiorganisaatio
   Ketkä ovat projektissa mukana ja miten se organisoidaan?
5) Projektiympäristö
   Minkälaisin menetelmin ja välinein työtä tehdään,
   onko resursseja varattu riittävästi?

Miten laatu saavutetaan?

- laatusuunnitelma
- testaussuunnitelma
- modulitestaus
- järjestelmätestaus
- testauspöytäkirjat
- hyväksymismenettelyt
- systemaattinen toiminta

Laadun kustannukset ja hyödyt

Motto: Mitä myöhemmin virhe havaitaan, sitä suuremmat korjauskustannukset

Laadun kustannukset

- Laadun kustannukset muodostuvat:
  - virheiden ennalta ehkäisystä
  - laadunvalvonnasta
  - virhekustannuksista
- Virheellisten tuotteiden osalta voidaan toimia:
  - tuote myydään halvemmalla --> tuotto vähenee
  - tuote korjataan --> työkustannukset nousevat
  - tuote heitetään pois --> panos menee hukkaan
  - tuote myydään virheettömänä --> PETOS!

Huonon laadun jäävuori