
OHJELMISTOJEN LAADUN JA KOON MITTAAMINEN

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

80

Mitat ja mittaus

- "You can't control what you can't measure" – Tom DeMarco, 1982.
- DeMarcon toteama on kaikkien mittausspesialistien motto: ilman mittausta ei ole ohjausta.
 - ◆ Väite on tietenkin liioiteltu. Kaikkia merkittäviä tietoja ei osata, voida tai haluta mitata. Mutta: **mittaamalla ohjaus helpottuu.**

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

81

Mitat ja Mittaus

- Yrityksen laatutavoitteet voidaan esittää erityisten indikaattorien tavoitearvojen avulla (haluttu taso)
- Esimerkiksi *asiakastyytyväisyyden ja -uskollisuuden mittaaminen* usealla indikaattorilla
 - ◆ Mieli-pidekyselyt
 - ◆ ”Net Promoter Score”
 - ◆ Häiriöilmoitusten ja tukipyyntöjen lukumäärän kehitys (suhteessa käyttäjämäärään)

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

82

Mitat ja Mittaus

- Vaikka mittauksia ei tehtäisikään täydessä laajuudessaan (tai tehdään kertaluonteisesti), mittausohjelma toimii *kommunikaation ja sitouttamisen* välineenä
 - ◆ Mittausohjelman toimeenpanoa tärkeämpää on saada kaikki ymmärtämään ja hyväksymään mitä tavoitellaan
- Huom! - mittauksen käyttö *palkitsemisen* tai *henkilökohtaisen suoriutumisen arvioinnin* perusteena kääntyy kuitenkin helposti itseään vastaan
 - ◆ Voi olla monta tapaa saada mittarit näyttämään haluttuja arvoja, eivätkä kaikki ole tarkoituksenmukaisia
 - ◆ Kehittäjät vierastavat henkilöön kohdistuvia mittauksia ja vaihtavat työpaikkaa

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

83

Mitta, mittaus, mittari

- *Mitta* tai *metriikka* (measure, metric) on luku tai symboli, joka karakterisoi kiinnostuksen kohteen (entity) tiettyä ominaisuutta (attribute)
- *Mittaus* (measurement) on prosessi, joka liittyy luvun tai symbolin jonkin tosimaailman olion ominaisuuteen sen kuvailemiseksi selvästi määritellyn säännön mukaan
- *Mittari* tarkoittaa yleensä mittaa, tai mittaa ja sen mittauksessa käytettyä menetelmää (~mittauksen määrätty suoritustapa)
 - ✦ Tekniikassa mittarilla tarkoitetaan yleisesti myös *mittalaitetta*, jolla tietty mittaus tehdään ja josta mittauksen tulos voidaan suoraan lukea

Katso myös: Mittaaminen, <http://fi.wikipedia.org/wiki/Mittaaminen>

Mitta, mittaus, mittari

- Mittarit voivat olla *suoria* (direct) tai *johdettuja* (indirect).
 - ✦ Suorien mittareiden mittaukset kertovat sellaisenaan kohteen tilasta. Esimerkiksi lämpötila on suora mittari.
 - ✦ Johdettujen mittareiden mittaukset saadaan yhden tai useamman suoran mittarin mittausten funktiona. Esimerkiksi kuukauden keskilämpötila on johdettu mittari.
- Mittarien avulla voidaan arvioida, ennustaa tai tilastoida mittajaan kannalta mielenkiintoisia kohteen ominaisuuksia.

Ohjelmistomittarit

- Ohjelmistotuotannossa on paljon suoria ja johdettuja mittareita, *ohjelmistomittareita* (software metrics).
- Ohjelmistomittareilla ja laatutekijöillä on yhteys. Useimpien laatutekijöiden sisältä voidaan löytää osatekijöitä, joiden toteutumista voidaan arvioida mittareilla.
- Kun laatutekijän osatekijöiden mittarit näyttävät, että osatekijät toteutuvat ohjelmistossa, laatutekijä toteutuu ohjelmistossa *ainakin osittain*.
 - ✦ Miksi vain osittain?
 - laatutekijä on yleensä enemmän kuin osatekijöidensä summa
 - kaikille osatekijöille ei ole varmoja mittareita

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen
laatu

86

LAATUMITTARIT

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen
laatu

87

Laatumittarit

- Laatumittarit mittaavat ohjelmistotuotteen tai ohjelmistoprosessin laatua.
 - ◆ Laatumittari on funktio, jonka syötteenä on ohjelmistoon liittyvä data ja tuloksena dataa kuvaava numeerinen arvo.
 - ◆ Laatumittarin numeerinen arvo (mittauksen tulos) kertoo, missä määrin mitattu data täyttää mittarilla kuvattavan laatuattribuutin (quality attribute)
 - Laatuattribuutti tarkoittaa jotakin laadun ominaisuutta. Esimerkiksi laatuominaisuudet ja -ominaisuudet (characteristics) ovat laatuattribuutteja
 - Arvon tulkinta vaatii usein sen suhteuttamista *kokeellisesti* tai *kokemusperäisesti* määritettyyn tavoitetasoon

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

88

"Onks kymppi paljon vai vähän?"



Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

89

Laatumittarien tarkoitus

- Laatumittarit ovat laadunvarmistuksen hyödyllisimpiä työkaluja. Niiden avulla
 - ✦ varmennetaan, että ohjelmistoprojektit etenevät oikein:
 - verrataan toteutunutta laatua ja laatutason vaihtelua suunniteltuun laatuun
 - verrataan toteutunutta aikataulua ja budjettia suunniteltuun aikatauluun ja budjettiin
 - ✦ tunnistetaan, milloin kehitys- tai ylläpitoprosessia pitää parantaa:
 - kerätään historiatietoa toteutuneista projekteista ja tuotteista
- Mittarit voivat olla
 - ✦ prosessikohtaisia: miten hyvin prosessi ja sen ilmentymä (projekti) vastaavat toisiaan
 - ✦ tuotekohtaisia: miten hyvin tuote ja sen laatutekijät vastaavat toisiaan.

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen
laatu

90

Laadun mittaus SQuaRE 2502n –standardien mukaan

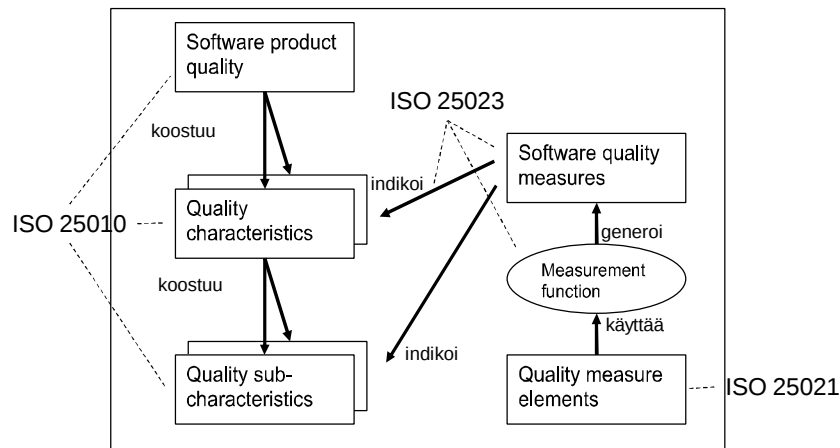
- Laatumalli 25010 määrittelee käytön laadun ja tuotelaadun piirteet, joista valitaan kullekin projektille relevantit piirteet
- 25020 ja 25021 luo puitteet ohjelmistomittareiden määrittelylle
- 25022 ja 25023 antavat suosituksia ja esimerkkejä, mitä mittareita kunkin laatupiirteen kohdalla kannattaa käyttää

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen
laatu

91

Laadun mittaus - tuotelaatu



Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

92

Laadun mittaus

- Ohjelmiston laatuun liittyviä mitattavia ominaisuuksia standardi kutsuu kvantifioitaviksi ominaisuuksiksi (property to quantify, ei ed. kuvassa)
- Ominaisuus kvantifioidaan käyttämällä mittausmenetelmää (measurement method), joka on toimenpiteiden sarja, joka liittää ominaisuuteen tietyn tyyppisen (measurement scale) arvon
- Saatua ominaisuuden arvoa kutsutaan perusmitaksi (quality measure element)

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

93

Laadun mittaus

- Perusmitat kytetään laatupiirteisiin johdettujen laatumittareiden kautta (software quality measures)
 - ◆ Kuhunkin laatumittariin liittyy mittausfunktio (measurement function) eli algoritmi, joka (yleensä) yhdistää useamman perusmitan yhdeksi laatumittarin arvoksi
 - ◆ Samaan laatupiirteeseen liittyy tyypillisesti useita laatumittareita, joiden avulla laatu kvantifioidaan
 - ◆ Laatumittarin määrittelystä ilmenee, mikä on peruste mittarin ja laatupiirteen yhteydelle

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

94

Tuotelaadun mittarit

- ISO 25023 määrittelee kullekin laatupiirteelle yhden tai useamman mittarin, ja antaa suosituksia niiden käytöstä
 - ◆ Aksellilla "Erittäin suositeltava – Suositeltava – Harkinnan mukaan"
 - ◆ Soveltuvuus ulkoisen ja/tai sisäisen laadun mittaamiseen
- Määrittelee myös perusmitat, joihin johdetut laatumittarit perustuvat sekä perusmittojen käyttämät mittayksiköt ja -asteikot (esim. koko, aika, työmäärä)
- Eri ohjelmistomittareita ja niiden suhdetta laatuun tarkastellaan lisää myöhemmin kurssilla!

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

95

Laatumittarien valinta

- Laatumittareita valittaessa määrä ei korvaa laatua. On parempi valita pieni joukko hyvin määriteltäviä mittareita kuin mitata kaikkea mahdollista
 - ✦ SQuaRE 25022, 25023 antavat suosituksia ja esimerkkejä
- Valittavat mittarit riippuvat siitä,
 - ✦ mikä on kiinnostavaa ja
 - ✦ mitä on ylipäänsä mahdollista mitata.
- Jotta mittarien käyttö olisi tehokasta, sen täytyy olla keskitettyä. Mittarit täytyy määritellä ylhäältä alaspäin (ensin tavoite, sitten siihen sopivat mittarit), sillä erilaisia mittareita on valtava määrä.
- Koska kaikkea ei voi eikä kannata mitata, valitut mittarit on perusteltava hyvin. Tähän voidaan käyttää *Goal Question Metric* -menetelmää (GQM, Basili & Rombach, 1988).

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

96

Goal - Question - Metric

- GQM^[1] perustuu näkemykseen, jonka mukaan järkevään mittaukseen yrityksen tarvitsee
 1. Määritellä yritys- ja projektitason tavoitteet
 2. Löytää ne yritys- projekti- ja tuotetason tiedot, jotka määrittävät tavoitteiden toteutumista yrityksen toiminnan tasolla
 3. Suunnitella kehys, jonka avulla kerätyt tiedot saadaan tulkittua kuvaamaan tavoitteita ja niiden toteutumista
- Eli:
 1. Mitä tietoa yritys tarvitsee?
 2. Miten tarpeet voidaan kvantifioida (esittää määrällisesti) tiedon keräämistä varten?
 3. Miten kerättyjä (numero-) tietoja pitää analysoida tavoitteiden saavuttamisen arvioimiseksi?
- GQM tarjoaa keinot määritellä kysymyksiin vastaava mittarijoukko.

[1] <http://www.cs.umd.edu/~mvz/handouts/gqm.pdf>

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

97

GQM-kolmitasomalli

● GQM on kolmitasoinen malli:

✦ Käsitetaso (GOAL)

- Selvitetään halutut tuote-, prosessi- ja resurssitavoitteet. Eli mittauksen kohde ja mitä siitä halutaan tietää tai muuttaa ja kenen näkökulmasta.

✦ Toimintataso (QUESTION)

- Tavoitteiden perusteella johdetaan joukko kysymyksiä, jotka kuvaavat kunkin tavoitteen toteutumista sitä luonnehtivien asiointilojen kautta
- Kukin kysymys luonnehtii mittauksen kohdetta laatuun liittyvän ongelman tai faktan kannalta ja tietyistä näkökulmista.

✦ Kvantitatiivinen taso (METRIC)

- Kuhunkin kysymykseen etsitään sellainen (objektiivinen tai subjektiivinen) mittari/mittarit, jonka avulla kysymykseen saadaan kvantitatiivisia vastauksia.

- GQM:lla ei kannata tehdä liian kunnianhimoisia suunnitelmia. Aluksi on parempi etsiä perusmittarit ydinkysymyksiin ja sen jälkeen vähitellen parantaa mittarikehystä lisäämällä kysymyksiä ja niihin liittyviä mittareita.

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

98

GQM malli

- GQM malli on hierarkkinen rakenne, joka alkaa tavoitteesta (goal)

- ✦ Tavoite määrittelee mittauksen tarkoituksen, mittauksen kohdeolion, itse mitattavan asian sekä näkökulman, josta mittaus tehdään

- Tavoite tarkennetaan kysymyksiksi (question), jotka jakavat mitattavan asian pääkomponentteihinsa

- Jokainen kysymys tarkennetaan sitten metriikoiksi (mittareiksi)

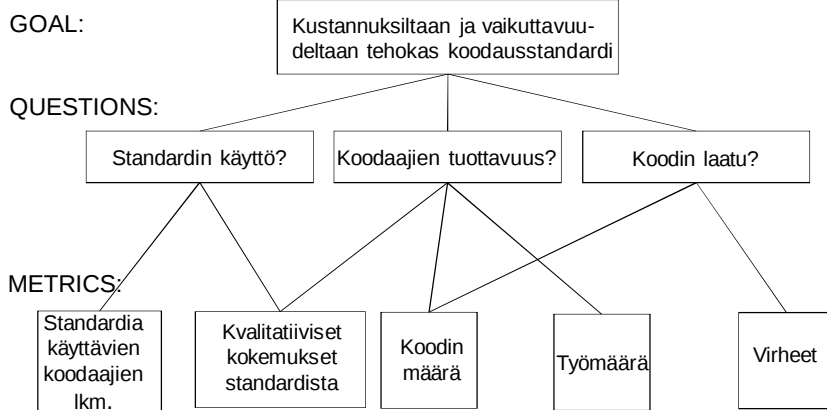
- ✦ Voivat olla objektiivisia tai subjektiivisia
- ✦ Samaa mittaria voidaan käyttää (osa-) vastauksena useampaan samaan tavoitteeseen liittyvään kysymykseen

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

99

GQM-esimerkki 1



Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

100

GQM-esimerkki 2

Tavoite	Tarkoitus	Parannetaan
	<i>Kohde (prosessi)</i>	muutospyyntöjen käsittelyn
	<i>Asia/ongelma</i>	ajallista täsmällisyyttä ja tehokkuutta
	<i>Näkökulma</i>	projektipäällikön näkökulmasta.
Kysymys 1		Mikä on nykyinen muutospyynnön käsittelynopeus?
Mittarit		• Muutospyynnön keskim. käsittelyaika • Keskipoikkeama • Käsittelyaikatavoitteen ylärajan ylittävien pyyntöjen osuus prosentteina
Kysymys 2		Tehostuuko käsittely?
Mittarit		• Nykyinen keskimäär. käsittelyaika prosentteina valitusta perustasosta • Projektipäällikön oma subj. arvio

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen laatu

101

GQM

- Usealla GQM mallilla voi olla yhteisiä kysymyksiä ja mittareita
 - ◆ Ohjaa siihen, että mittauksen suorituksessa voidaan ottaa huomioon eri näkökulmat mitattavaan asiaan
 - ◆ Sama mittaus voi tuottaa erityyppisiä arvoja näkökulmasta riippuen

KOON MITTARIT

Ohjelmiston kokomittarit

- Huomattavassa määrässä ohjelmiston laatumittareita mitataan niiden osana ohjelmiston kokoa. Vaikka erilaisia kokomittareita on lukemattomia, kaksi on ylitse muiden:
 - ✦ KLOC (Kilos Lines Of Code): Kuinka monta tuhatta koodiriviä on ohjelmistossa.
 - ✦ FP (Function Points): Kuinka paljon toiminnallisuutta on ohjelmistossa.
- Muita kokomittareita:
 - ✦ Montako lausetta on ohjelmistossa (KLOC-variantti).
 - ✦ Montako luokkaa/metodia/funktiota on ohjelmistossa.
 - ✦ Montako *riippumatonta polkua* (independent path) on ohjelmistossa tai (yleensä) sen osassa.
 - Riippumaton polku on sellainen suoritusreitti ohjelmistossa, että sitä ei voi saada yhdistämällä muista riippumattomista poluista.

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen
laatu

104

KLOC

- KLOC on yleisin ohjelmiston kokomittari. Sillä mitataan ohjelmiston kokoa koodiriveinä joko kommenttien kanssa tai ilman niitä.
- KLOC on selkeä mittari, joka kuvaa koon lisäksi yllättävän hyvin kompleksisuutta: mitä isompi ohjelmisto/ohjelma/luokka/metodi tms. on, sitä mutkikkaampi se yleensä on (ja esimerkiksi sitä hankalampi testata ja ylläpitää).
- KLOC:n heikkous on sen ohjelmointikieliriippuvuus. Eri ohjelmointikielillä toteutetuista ohjelmistoista lasketut KLOC-luvut eivät ole keskenään verrannollisia.
- Myös koodaustyyli vaikuttavat jonkin verran koodirivien määrään, mutta isoilla ohjelmistoilla erojen suhde koodirivien kokonaismäärään on melko pieni.
 - ✦ KLOC:n sijaan kannattaa laskea esimerkiksi puolipisteitä (so. lauseita), jos koodirivin pituuden vaihtelu osoittautuu ongelmaksi.

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen
laatu

105

KLOC-ongelmia

- Koska KLOC riippuu käytetystä ohjelmointikielestä ja koodaustavasta, KLOC-arvot ovat monen, laadun kannalta vähemmän tärkeän muuttujan summa.
- Lisäksi KLOC:lla on ikävä ominaisuus: se on nimensä mukaisesti koodirivien määrä, joten **sitä ei voida laskea, ennen kuin koodi on kirjoitettu.**
- Miten lasketaan kirjastoista/kehyksistä *uudelleenkäytetyt* koodirivit??
 - ✦ Lähdekoodia ei edes usein saatavissa
- KLOC-arvoja voidaan arvioida, mutta arviot ovat summittaisia. Olisi parempi, jos jo määrittely- tai suunnitteluvaiheessa ohjelman koolle voisi antaa luotettavan arvion.
- Määrittelystä laskettava kokoarvio on mahdollista *toimintopisteillä* (function points). Niiden avulla jo melko yleisistä määrittelyistä voidaan arvioida, miten paljon toiminnallisuutta ohjelmistossa on.
 - ✦ Arviot tarkentuvat projektin edetessä ja ohjelmiston vaiheittain valmistuessa

Kevät 2015

Ohjelmistoprosessit ja ohjelmistojen
laatu

106