

PROSESSIN MITTAREITA

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

136

Prosessin mittareita

- Kokomittojen määrittelyn jälkeen on helppo määritellä lukuisia prosessin tehokkuuden mittareita, joissa mitattavaa suuretta verrataan ohjelmiston kokoon. Tässä muutama:
 - ◆ Vikatiheys $DD = NoD / KLOC$ (tai FP), missä NoD on tunnettujen* vikojen määrä (ohjelman kehityksen ja/tai käytön aikana löydetty viat).
 - ◆ Painotettu vikatiheys: $WDD = WNoD / KLOC$, missä $WNoD$ on tunnettujen vikojen määrä painotettuna niiden vakavuusasteella.
 - ◆ Vikojen poiston tehokkuus: $DRE = \text{Kehitysaikana korjatut viat} / NoD$, missä NoD on nyt kehitysaikana havaittujen ja tietyn *käyttöjakson* aikana havaittujen vikojen määrä (esim. vuodessa).
 - ◆ Kehitystyön tuottavuus: $DevP = DevH / KLOC$, missä $DevH$ on projektiin käytetty kokonaistyöaika.
 - ◆ Kehitystyön tuottavuus toimintopisteillä: $FDevP = DevH / FP$.
 - ◆ Koodin uudelleenkäyttö: $CRe = ReKLOC / KLOC$, missä $ReKLOC$ on uudelleenkäytettyjen koodirivien määrä / 1000.
- GQM-menetelmällä tai vastaavalla voidaan etsiä yrityksen kannalta hyödyllisimmät mittarit.

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

137

(*) tunnettujen vikojen (defect) lisäksi ohjelmistoon jää aina myös tuntemattomia vikoja, joita ei (vielä) ole havaittu

Prosessin mittareita: esimerkki

- Olkoon ohjelmiston koko 40.000 riviä (KLOC=40). Siitä on löydetty projektin aikana 70 ohjelmakoodivikaa, joista 11 on erittäin vakavia, 17 melko vakavia ja 42 vähäpätöisiä.
- Määritellään vikojen vakavuusasteille seuraavat painokertoimet:
vähäpätöinen: 1, melko vakava: 3, erittäin vakava: 9.
 - ◆ (Kehityksen aikainen) Vikatiheys: $DD = NoD / KLOC = 70 / 40 = 1,75$.
 - ◆ Painotettu vikatiheys: $WDD = CC / KLOC = (9 \cdot 11 + 3 \cdot 17 + 1 \cdot 42) / 40 = 192 / 40 = 4,8$.
- Olkoot mittareilla laatukynnysarvot $NoD \leq 2$ ja $WDD \leq 4$. Tällöin NoD ei ole viite ohjelmiston huonosta laadusta, mutta WDD on (liikaa erittäin vakavia vikoja).

Vikojen määriin perustuva mittaus

- Kuten jo kurssin alussa todettiin, vikojen ja häiriöiden vähäinen määrä on intuitiivinen ja yleisesti käytetty laadun mittari
- Vikoihin perustuvissa mittareissa on kuitenkin pidettävä mielessä seuraavat asiat:
 1. Kaikki viat eivät johda häiriöihin; vika on ohjelman koodin ominaisuus ja häiriö ohjelman suorituksen ominaisuus
 - Niitä havaitaan eri keinoin ja usein eri elinkaaren vaiheissa
 - Pitää olla tarkkana, että kaikki osapuolet ymmärtävät mitä asiaa täsmälleen mitataan

Vikojen määriin perustuva mittaus

2. Tarkoitetaanko vikatiheydellä vikojen määrää suhteutettuna ohjelmiston kokoon vai tiettyyn *aikaikkunaan*, jolloin vikoja havaitaan? Usein käytetään termiä "defect rate" epätasaisesti edellisessä merkityksessä.
3. On tärkeää ymmärtää, miten ohjelmiston koko mitataan, jos vertaillaan eri *projektien/ohjelmistojen* vikatiheyksiä toisiinsa (KLOC variantit, FP)
4. Vikatiheyden arvoon vaikuttaa paitsi vikojen määrä, myös vikojen löytämiseen käytettyjen *resurssien määrä ja laatu* sekä *raportoinnin tarkkuus* (voi olla vaikea saada vertailukelpoisia tietoja eri yrityksistä)

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

140

Vikojen määriin perustuva mittaus

5. Vaikka "jäännösvikojen" (residual defects) määrästä olisikin hyvä käsitys, on johtopäätöksissä oltava varovainen
 - Vian vakavuutta voi olla vaikea arvioida etukäteen
 - Käyttäjien erilaisista käyttötavoista johtuen voi olla vaikea ennustaa, mitkä viat voivat todella johtaa häiriöihin ja kuinka usein
 - Ennakko-odotukset uusien vikojen löytymisen ajallisesta jakaumasta saattavat muuttua itseään toteuttavaksi harhaksi

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

141

Muita ”vikamittareita”

- System spoilage = $\frac{\text{Time to fix post-release defects}}{\text{Total system development time}}$
- Kehittäjäorganisaation löytämien vikojen kumulatiivinen määrä
- Asiakkaiden löytämien vikojen kumulatiivinen määrä
- Löytyneiden vakavien vikojen kokonaismäärä
- Vakavan vian korjaamiseen kulunut aika keskimäärin
- Eri kehitysvaiheissa (vian syntyvaihe) ja testaus-tasoilla löydettyt eri vakavuusasteiset viat / KLOC
- Kypsyystason (maturity) päättely vikojen löytymistahdin (defect discovery rate) perusteella

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

142

LAADUNVARMISTUS

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

143

Projektinhallinnan laadunvarmistus

- Projektinhallinnan laadunvarmistus tukee ohjelmistoprojektien ohjaus- ja ylläpitotehtäviä.
- Projektinhallinnan laadunvarmistustehtäviin kuuluvat seuraavat:
 - ◆ projektien etenemisen seuranta ja ohjaus
 - ◆ projektien laadun mittaus ja arviointi
 - ◆ laadun kustannusten mittaus ja arviointi

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

144

Projektien etenemisen seuranta ja ohjaus

- Projektien etenemisen seuranta ja ohjaus on projektinhallinnan laadunvarmistuksen tärkein tehtävä. Sen avulla varmistetaan, että:
 - ◆ projektit pysyvät aikataulussa ja budjetissa
 - ◆ projektien riskienhallinta toimii
 - ◆ projektit reagoivat toteutuneisiin riskeihin ja muuttuneisiin reunaehdoihin oikein.
- On tärkeää, että projekteja seurataan ulkopuolelta. Omalle työlle on helppo tulla sokeaksi. Ulkopuolinen taho näkee työn etenemisen toisella tavalla kuin projektityöntekijä.
- Seuranta ja ohjaus eivät kuitenkaan ole projektista riippumattomia. Projekti raportoi omasta etenemisestään laadunvarmistuksesta vastaaville henkilöille, ja epäselvissä tilanteissa laadunvarmistuksesta kysytään tietoja projektilta.

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

145

Projektien laadun mittaus ja arviointi

- Laadunvarmistuksessa projektien laatua arvioidaan sopivilla projektia mittaavilla metriikoilla.
 - ◆ Hallittava mittauksiin mahdollisesti liittyvien väärinkäytösten ja harhojen riskit
- Metriikat vaihtelevat yrityksittäin. Seuraavassa on muutama yleinen:
 - ◆ tuottavuudelle koodirivien määrä / kuukausi
 - ◆ testaustehokkuudelle löydettyjen virheiden määrä / 1000 koodiriviä (kloc)
 - ◆ työtehokkuudelle tuottava työaika / kokonaistyöaika

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

146

Laadun kustannusten mittaus ja arviointi

- Projekteissa arvioidaan myös laadunvarmistuksen hyöty-kustannus-suhdetta: paljonko laadunvarmistus kuluttaa ja paljonko sillä säästetään.
- Jos laadunvarmistus säästää enemmän kuin kuluttaa, se on kannattavaa.
- Optimaalisen laadunvarmistustason löytäminen on optimointitehtävä. Tiettyyn tasoon asti laadunvarmistus vähentää kustannuksia, mutta ei rajattomasti.

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

147

Standardointi

- Ulkoisten standardien käyttö on yksi tapa yhtenäistää ohjelmistoprojekteja ja siten saada niistä ulos tasalaatuisempia tuotteita.
- Ulkoiset standardit ovat yleisiä ja yleensä huolellisesti testattuja. Ne sopivat useimpiin organisaatioihin jossain määrin mutta eivät juuri mihinkään organisaatioon täydellisesti sellaisenaan.
- Käytännössä laatustandardi täytyy sovittaa omaan yritykseen. Tämä voi olla helppoa tai vaikeaa riippuen siitä, minkälainen yritys on kyseessä, minkälaisia prosesseja seurataan ja minkälaisia ohjelmistoja valmistetaan.

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

148

Kansainvälisiä laatustandardeja

- Laadunvarmistuksen kannalta mielenkiintoisia ovat laadunhallinnan standardit ja prosessistandardit.
 - ◆ Laadunhallinnan standardit määrittelevät, mitä laadukkaaseen ohjelmistokehitystyöhön vaaditaan. Sen sijaan ne eivät määrittele, miten yritys toteuttaa ko. vaatimukset. Tällaisia standardeja ovat:
 - SEI-CMMI: Software Engineering Institute - Capability Maturity Model Integration
 - ISO-9001:2000: International Standards Organization – Quality Management Systems: Requirements
 - ISO 15504 (SPICE): Software Process Improvement and Capability determination
 - ◆ Prosessistandardit määrittelevät, miten kehitystyötä tai sen osavaihetta tehdään, ja sen kautta määrittelevät, mitä näin saavutetaan. Tällaisia standardeja ovat
 - IEEE 1012: Institute of Electrical and Electronic Engineers - IEEE Standard for Software Verification and Validation
 - ISO/IEC 29119: Software Testing Standard
 - ISO/IEC 12207: ISO/International Electrotechnical Commission – Software Life Cycle Processes

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

149

Laatustandardit

- *Laatustandardi* (quality standard) tarkoittaa yleensä kansallista tai kansainvälistä laatuun liittyvää ohjeistoa, joka on käynyt läpi usean raakaversion ennen hyväksymistä.
- Laatustandardit helpottavat laadunvarmistuksen suunnittelua ja hallintaa, sillä ne on laadittu huolella sellaisiksi, että ne sopivat sellaisenaan tai pienillä muutoksilla hyvin erilaisten yritysten laadunvarmistuksen rungoksi.
- Laatustandardeja ei ole pakko käyttää. Yritys voi kehittää itselleen sopivimman laadunvarmistusprosessin ja käyttää sitä menestyksekkäästi.
- Toisaalta hyvän prosessin kehittäminen on työlästä ja kallista, joten vähintään prosessiin kannattaa lainata oman yrityksen kannalta hyviä ideoita olemassa olevista standardeista.

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

150

Laatustandardien edut

- ◆ Yritys saa käyttöönsä kehittyneimmät kehitys- ja ylläpitomenetelmät
 - Laatustandardit on testattu erittäin huolellisesti ennen kuin niistä on tehty viralliset versiot
- ◆ Standardit parantavat projektiryhmien keskinäistä ymmärrystä ja koordinaatiota
 - Yhteinen standardi tarkoittaa yhteistä kieltä, mikä parantaa kommunikaatiota
- ◆ Yhteistyö ohjelmiston kehittäjien ja ulkoisten sidosryhmien välillä paranee
 - Laatustandardi on julkinen, joten ulkoiset sidosryhmät voivat sen kautta viitata projektin tai tuotteen laatuun
- ◆ Yhteistyö asiakkaiden ja alihankkijoiden kanssa paranee
 - Laatustandardi on usein asiakkaalle vakuuttavampi dokumentti kuin itse laadittu ohjeisto
 - Laatustandardi määrittelee (tai sen pitäisi määritellä) myös alihankkijoiden työn laadun

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

151

Laatustandardien ongelmat

- ◆ Standardi voi olla yrityksen kannalta liian yleisellä tasolla
 - Standardin pitää sopia eri tyyppisiin yrityksiin ja projekteihin melko hyvin, mutta useimpiin yrityksiin ja projekteihin se ei sovi sellaisenaan täydellisesti
- ◆ Standardi voi olla yritykselle liian raskas
 - Varsinkin pienten yritysten voi olla vaikeaa toteuttaa haluttua laatustandardia, koska niissä ei ole tarpeeksi resursseja kaikkien laatustandardissa määriteltyjen tehtävien toteuttamiseen
- ◆ Standardi ei välttämättä ole yhteensopiva yrityksen nykyisten toimintatapojen kanssa
 - Laatustandardissa määritellään melko tarkasti, mitä kaikkea laadunvarmistukselta vaaditaan, mutta yrityksessä käytetyt toimivat menetelmät voivat olla sellaisia, että ne eivät taivu laatustandardiin ilman perinpohjaista muutosta sekä yritys- että projektitasolla