

Metriikat käytänteiden tukena ohjelmiston laadun arvioimisessa

Ohjelmiston laatua voidaan tutkia mekaanisilla metriikoilla, joiden tehtävänä on arvioida ohjelmiston komponentteja sekä pystyä havaitsemaan ohjelmiston kriittiset osa-alueet. Näitä metriikoita ovat mm. koodikirnu, verkkoanalyysi, testikattavuus ja mutaatiotestaus. Ohjelmiston laadun arvioimisessa on hyvä käyttää useita eri metriikoita, koska yksittäiset metriikat paljastavat vain osatotuuden ohjelmiston laadusta. Laadun varmistaminen tulee kohdentaa metriikoiden havaitsemiin virhealttiisiin kohteisiin ohjelmistossa.

Laadun varmistamista rajaavat ohjelmistokehityksessä henkilöt, aika ja raha. Päätökset tehdään usein johtajien odotusten mukaan, ja näin ollen he joutuvat arvioimaan laatua puutteellisin tiedoin. Kriittiseksi osaksi muodostuvat siksi kehittäjien taitojen lisäksi johtajien taidot.

Ohjelmiston laadukkuus painottuu vahvasti asiakkaisten tarpeiden täyttymykseen. Laadullisten kriteereiden osa-alueita on neljänlaisia: laatumalli, ulkoinen ja sisäinen malli käyttölaadulliset metriikat. Metriikat tarjoavat yhden tehokkaan keinon ohjelmistojen laadun arviointiin. Monimutkaisuusmetriikat ovat muodostettu esimerkiksi koodirivien, metodien ja muuttujien lukumäärästä.

Koodikirnulla arvioidaan ohjelmiston muutoksien vaikutusta ohjelmiston virheherkkyyteen. Koodi, joka muuttuu useasti ennen julkaisua on virheherkempää kuin koodi, joka muuttuu vähemmän saman ajanjakson aikana. Tutkimukset ovat osoittaneet, että ehdottomat mittayksiköt, kuten pelkkä koodirivien summa, ovat huonoja ohjelmiston laadullisia ennusteita.

Verkkoanalyysillä tutkitaan ohjelmiston komponenttien riippuvuuksien vaikutusta ohjelmiston virhealttiuteen. Järjestelmän riippuvuudet voidaan esittää matalan tason verkkoina, jossa komponenttien keskinäiset suhteet paljastuvat. Näille riippuvuusverkoille voidaan suorittaa verkkoanalyysi ohjelmiston virheherkkyyden arvioimiseksi. Mitä suurempi joukko verkossa olevia komponentteja riippuu keskenään toisistaan, sitä suurempi on näiden komponenttien todennäköinen virheherkkyys. Verkkoanalyysillä voidaan löytää jopa kaksi kertaa enemmän kriittisiä komponentteja kuin pelkillä monimutkaisilla metriikoilla.

Testikattavuuden lajeja on useita. Yksinkertaisista luokka-, funktio-, metodi- ja käskykattavuuksista kehittyneisiin haara- ja polkukattavuuksiin. Testikattavuus on yksi keino parantaa ja selvittää ohjelmiston laatua. Tutkimuksissa on havaittu, että suurempaa testikattavuutta seuraa pienempi määrä julkaisun jälkeisiä virheilmoituksia. On vaikeaa tietää, kuinka suuri osa virheistä havaitsi testikattavuus, koska suurin osa virheistä paikataan samalla kun ohjelmoidaan yksikkötestauksen aikana. Testikattavuuden ongelmaksi muodostuu se, että se ei arvioi testien laatua. Se tarkastelee vain mitä osia ohjelmiston lähdekoodista on katettu testitapauksilla.

Testien laadun arvioimiseen on ehdotettu mutaatiotestausta. Mutaatiotestaus arvioi testien sopivuutta niiden kattamaan lähdekoodiin simuloimalla yleisempiä virheitä joita kehittäjät tekevät. Mutaatiotestaus muodostaa virheellisiä mutanttiversioita ohjelmistosta, jotka ajetaan ohjelmiston testitapausten läpi ja samalla tutkitaan havaitsevatko testit mutanttien väärän toiminnallisuuden.

Ketterän kehityksen idea on mahdollistaa muutokset kehitystyössä. Kehitystyötä tehdään iteratiivisesti pienissä palasissa. Ketterän kehityksen on todettu vähentävän kehitykseen kuluva aikaa pitkällä tähtäimellä. XP ja Scrum ovat nykyään ketterän kehityksen käytetyimmät prosessit, jotka tukevat toisiaan.