Metriikat käytänteiden tukena ohjelmiston laadun arvioimisessa, Kasper Hirvikoski

Artikkelissa esitellään lukuisista metriikoista muutama hyödylliseksi havaittu: koodikirnu, vekkoanalyysi, testikattavuus ja mutaatiotestaus.

Koodikirnulla arvioidaan ohjelmiston muutosten vaikutusta ohjelmiston virheherkkyyteen. Lähdeaineisto saadaan versionhallintajärjestelmästä. Koodikirnu mittaa ohjelmiston komponentteihin kohdistuvien muutosten määrää tietyn ajanjakson aikana. Koodikirnu koostuu joukosta ehdottomia ja suhteellisia mittoja, joiden tuloksia verrataan keskenään ohjelmiston virheherkkyyden analysoinnissa. Suhteelliset koodikirnumitat ennustavat paremmin virhetiheyttä.

Verkkoanalyysillä paikallistetaan ohjelmiston kriittiset komponentit ja tutkitaan niiden välisten riippuvuuksien vaikutusta ohjelmiston virhealttiuteen. Tietoriippuvuus on määrittelyiden ja arvojen välinen yhteys ja kutsuriippuvuus on funktio- ja metodimäärittelyiden ja niitä kutsuvien paikkojen välinen yhteys.

Verkkomittoja ovat egoverkot, globaaloverkot, rakenteelliset puutteet ja keskeisyys.

Testien kykyä havaita mahdollisia virheitä ohjelmistosta arvioidaan testikattavuudella. Tällöin tarkastellaan vain, mitä osia ohjelmiston lähdekoodista on katettu testitapauksilla eikä arvioida testien laatua. Funktio- ja metodikattavuudella kartoitetaan testien kattavuutta yhden toiminnallisuuden osalta ja luokkakattavuudella kartoitetaan kokonaisuuden testien kattavuutta. Testikattavuuden suhteen pitää varmistaa, että varsinaiset testit testaavat oikeita asioita.

Mutaatiotestauksella analysoidaan lähdekoodin testien tehokkuutta ja laatua simuloimalla yleisimpiä kehittäjien tekemiä virheitä. Mutaatiotestaus muodostaa virheellisiä mutanttiversioita ohjelmistosta ja tutkii, miten hyvin testaus havaitsee muutokset. Mutaatiotestausta voidaan käyttää yksikkö-, integraatio ja määritelmätason testeihin. Tärkeintä on valita oleellisimmat tapaukset, joista mutantteja tehdään.

Metriikat tarjoavat keinon parantaa ohjelmiston laatua, mutta eivät kuitenkaan välttämättä ota huomioon virheiden inhimillistä tekijää, ohjelmiston kehittäjää.