Automaatiojärjestelmät 1

Ydinvoima-automaatio ja laadunvarmistus: lukutehtävä

Tehtävänanto

Pisteytys

Yhteensä 10p (ydinvoima 8p ja testaus ja laadunvarmistus 2p).

Ydinvoima (8p)

Hae STUKin julkaisut sivulta Säteily- ja ydinturvallisuus –kirjasarjan kirja Ydinturvallisuus ja sieltä luku

”Ydinvoimalaitostekniikan perusteita”.

https://www.stuk.fi/julkaisut/sateily-ja-ydinturvallisuus-kirjasarja/ydinturvallisuus

Mieti vastaukset seuraaviin kysymyksiin ja esitä ne suullisesti harjoitusten vetäjälle. Esityksessä saa olla mukana omat muistiinpanot. *Useampi ryhmä voi suorittaa esityksensä yhtä aikaa (kaikilta kuitenkin kysytään jotain).* Vastaukset löytyy luvuista 2.1-2.3. Luvuista 2.2 – 2.3 riittää katsoa ne kohdat jotka liittyy kiehutusvesireaktorin.

1. Mikä on kasvutekijä?
2. Mitä tarkoittaa, että reaktori on kriittinen, alikriittinen tai ylikriittinen?
3. Miten säätösauvojen työntäminen sisään vaikuttaa kriittisyyteen?
4. Miten polttoaineen, moderaattorin ja jäähdytteen lämpötilan reaktiivisuuskerroin vaikuttaa reaktorin tehon muutokseen? Millä kertoimien arvoilla takaisinkytkennät vaikuttavat reaktorin tehoa stabiloivasti?
5. Minkälaisissa tilanteissa reaktorin säätöjärjestelmillä muutetaan reaktorin tehoa? Miten nopeista säätötehtävistä on kysymys? (Numeerisia tietoja ei edellytetä.)
6. Kuvaa hyvin lyhyesti kiehutusvesireaktorin toimintaperiaate jäähdytteen ja moderaattorin kierron näkökulmasta? Käytä tukena kaaviokuvaa kiehutusvesireaktorista
7. Mitkä ovat kevytvesireaktorin tärkeimmät turvallisuustoiminnot ja niitä toteuttavat turvallisuusjärjestelmät?
8. Luettele kiehutusvesireaktorin automaatiojärjestelmän keskeiset osajärjestelmät ja mittaukset?

Ohjelmistojen testaus ja laadunvarmistus

Teoria

Ohjelmistojen testaus ja laadunvarmistus ovat laaja aihe, josta tietotekniikkaopiskelijoille on kokonainen kurssi. Tässä keskitytään kaikkein olennaisimpaan sisältöön, jotta keskeisimmät käsitteet tulisivat tutuksi ja jotta opiskelijoilla olisi jotain käytännössä toimivia tekniikoita testaamiseen.

Olet varmasti jonkun verran jo testaillut ohjelmistoja, jos ei muualla niin ohjelmoinnin peruskurssilla. Milloin olet mielestäsi testannut riittävästi? Miten suunnittelet testin?

Testauksen on käytännöllinen ja hyvin yleinen tapa suorittaa laadunvarmistusta ja se tehdään ajamalla ohjelmistoa. On myös monenlaisia tieteellisempiä tapoja analysoida ohjelmistoa, jotta varmistutaan että sillä on joku haluttu ominaisuus, mutta niiden käyttö yliopistomaailman ulkopuolella ja erityisesti automaatioalan yrityksissä on vähäistä.

Testauksen perusongelma on, että missä tahansa kaupallisen tasoisessa ohjelmistossa on lukematon määrä mahdollisia syötteitä ja niiden mukaisia sekvenssejä. Kaikkea ei voida siksi kokeilla, vaan pitää suunnitella rajallinen määrä testejä, jotka testaavat mahdollisimman suuren tai relevantin osan mahdollisista syötteistä.

Testauksen tarkoitus on varmistaa, että ohjelmisto toimii *vaatimusten (requirements)* mukaisesti. Jos vaatimuksia ei kirjoitettu auki ennen ohjelmointia, niin systemaattinen testaus ei ole mahdollista. Esimerkkejä vaatimuksista tulee tehtävässä ”Ydinvoima-automaatio: ohjelmointi ja laadunvarmistus”. Vaatimusmäärittely ei ole vain akateemisen piirin harrastusta, vaan sen merkitys on laajasti ymmärretty teollisuudessa.

Jos testit suunnitellaan vaatimusten perusteella ilman tietoa ohjelmistosuunnitteluun liittyvistä ratkaisuista, puhutaan *mustalaatikkotestauksesta (black-box testing)* ja *toiminnallisesta testauksesta (functional testing)*. Muussa tapauksessa puhutaan *valkolaatikkotestauksesta (white box testing)*. Testauksessa pyritään mahdollisimman korkeaan kattavuuteen, joka voidaan määritellä esimerkiksi niin että kuinka monta prosentti koodiriveistä tulee suoritettua vähintään yhdessä testissä tai kuinka suuri osa ehtolauseiden haaroista tulee valittua vähintään yhdessä testissä.

Valkolaatikkotestauksella voidaan nostaa kattavuutta. Testit voivat mennä rikki, kun ohjelmaa muutetaan, vaikka vaatimukset eivät muuttuisi. Toiminnallisessa testauksessa puolestaan halutaan varmistaa, että kaikkiin vaatimuksiin on olemassa ainakin yksi testi. Keinona tässä toimii *jäljitettävyys (traceability)*, eli testeistä on joku linkki niihin vaatimuksiin, joita testataan. Jotkut IT alalla käytetyt työkalut antavat työkalutukea jäljitettävyyteen. Lowtech ratkaisu on tehdä linkityksiä käsin Word tai Excel muodossa oleviin vaatimuslistoihin ja testisuunnitelmiin, mutta tätä on hankalaa ylläpitää, kun dokumentteja muutetaan jatkuvasti.

Kaikki yllämainitut testaukseen liittyvät asiat ovat erityisen tärkeitä *turvallisuuskriittisessä (safety critical)* automaatiossa. Viranomaishyväksynnän saamiseksi (esim EU:n konedirektiivi) ja mahdollisista haavereista johtuvien oikeudenkäyntien hoitamiseksi, automaatiotuotteen valmistajan tulee voida dokumentaation voimalla osoittaa, että laadunvarmistustekniikoita on käytetty systemaattisesti. Jos kysymyksessä ei ole

turvallisuuskriittinen tuote, on laadunvarmistukseen liittyvät määritykset kytkettävä siihen, miten paljon virheellisen tuotteen laskeminen markkinoille tulee haittaamaan liiketoimintaa. Esimerkiksi buginen kuluttajatuote haittaa liiketoimintaa palautetuista tuotteista menetettyinä katteina ja mainehallinnan kautta.

Turvallisuuskriittisen järjestelmän kehitys ja sertifiointi ovat kallista. Osin tästä syystä, automaatioalalla järjestelmä voidaan jakaa erikseen käyttöautomaatioon ja turva-automaatioon. Turva-automaatio suorittaa turvatoimintoja, joissa luotettavilla ja mahdollisesti kahdennetuilla antureilla havaitaan vikatilanteita ja ajetaan prosessi tarvittaessa turvalliseen tilaan. Prosessiautomaatiossa ja erityisesti ydinvoima-alalla on normaalia käyttää erillistä turva-automaatiota. Vastaavasti koneautomaatiossa on yleistä, että ei haluta investoida erilliseen turva-automaatioon, joten tavallisen automaation osalta laadunvarmistuksen osuus ja huolellinen dokumentointi korostuvat.

Vaikka laadunvarmistus on iso osa turvallisuutta, turvallisuus on muutakin. Siinä kaikki alkaa riskinarvioinnista, jossa kartoitetaan riskit, niiden todennäköisyys ja vakavuus ja näiden perusteella määritellään toimenpiteet, joihin riskin varalta mahdollisesti ryhdytään. Tästä tulee enemmän ’Hissi’ tehtävän yhteydessä.

Yllä kuvattiin huomattava määrä tärkeitä periaatteita, mutta paras tapa sisäistää asia on käytännön tekemisen kautta, joten vaatimusmäärittelyn ja testauksen parissa jatketaan ”Ydinvoima-automaatio: ohjelmointi ja laadunvarmistus” tehtävän yhteydessä.

Kysymykset (2p – riittää että keskustelee aktiivisesti)

- Millä kursseilla olet tähän mennessä tehnyt ohjelmointitehtäviä? Oliko vaatimukset annettu kokonaan kurssin puolesta? Oliko ne kattavasti ja yksikäsitteisesti määritelty? Onko sinulle opetettu, että miten toimitaan tilanteessa, jossa asiakas ei ole kykeneväinen tai halukas määrittelemään vaatimuksia kattavasti tai yksikäsitteisesti?
- Millä kursseilla olet tähän mennessä tehnyt testausta? Missä määrin suunnittelit testejä? Millä perusteella päädyit tiettyihin testeihin?