

Testaussuunnitelma

Viski-ryhmä

Helsinki 31.8.2006

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Esa Elovaara
Suvi Hiltunen
Tomi Jylhä-Ollila
Riku Louhimo
Samuli Sairanen
Juho Vuori

Asiakas

CSC / Aleksi Kallio

Johtoryhmä

Juha Taina
Jaakko Saaristo

Kotisivu

<http://www.cs.helsinki.fi/group/viski>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	12.7	Runko lisätty svn-repoon

Sisältö

1	Johdanto	1
1.1	Testauksen tavoitteet ja rajat	1
2	Testausvaiheet	1
2.1	Yksikkötestaus	1
2.2	Integroititestaus	1
2.3	Järjestelmätestaus	2
3	Muu testaus	2
3.1	Volyymitestaus	2
4	Testausaikataulu	2

1 Johdanto

Tämä on viski-projektin testaus suunnitelma. Dokumentissa käsitellään testauksen eri vaiheita ja työkaluja. Projektissa tehdään JFreeChart-kirjastoa apuna käyttäviä visualisointeja tieteellistä analyysia varten.

1.1 Testauksen tavoitteet ja rajat

Koska viski-projekti toteuttaa laajennuksen JFreeChart-kirjastoon, varmistetaan lisäyksen toimivuus JFreeChart-kirjaston osana. Itse JFreeChart-kirjastoon yksikkötestausta ei kohdisteta, koska se on jo testattu olemassa olevilla yksikkötesteillä. Integroititestausvaiheessa visualisointeja testataan JFreeChart-kirjaston osana, jotta varmistutaan, että JFreeChart-kirjaston rajapintoja käytetään oikein.

Testausta valvoo projektin testausvastaava, Samuli Sairanen.

2 Testausvaiheet

Testaus noudattaa Ohjelmistotuotanto-kurssilla esitellyn V-mallin mukaista toimintamallia.

2.1 Yksikkötestaus

Jokaista luokkaa varten toteutetaan yksikkötestejä, joiden tarkoituksena on testata yksittäisen luokan toimintaa ja vikasietoisuutta. Yksikkötestin tavoitteena on käydä läpi kaikki ohjelman koodirivit (lausekattavuus).

Testien suunnittelussa käytetään arvoalueanalyysia. Testattaessa int-parametreja tulee käyttää parametrien arvoja minimi -1 , maksimi $+1$, maksimi ja minimi. Olio-parametreja testattaessa riittää null ja olemassa oleva olio.

Yksikkötestit kirjoitetaan käyttäen apuna JUnit-kirjastoa, joka on Javaa varten kehitetty testausympäristö.

Testauksen kattavuus varmistetaan käyttämällä Cobertura-ohjelmistoa, joka kertoo ajettujen yksikkötestien kattavuudesta. Tavoitteena on 100 %:n lausekattavuus.

Yksikkötestit sijoitetaan luokkien hakemistoihin tehtäviin junit-nimisiin alihakemistoihin. Yksikkötestien raportit tuottaa Cobertura-ohjelmisto.

2.2 Integroititestaus

Integroititestausvaiheessa kootaan yksittäiset luokat osajärjestelmiksi ja testataan osajärjestelmiä kokonaisuutena. Osajärjestelmän osat on kertaalleen yksikkötestattu, joten

integroititestausten huoleksi jää rajapintojen välinen toiminta. Integroititestausta toteutetaan ”bottom-up” -menetelmällä, jossa pienimmät luokat yhdistetään isommiksi kokonaisuuksiksi ja tätä jatketaan etenemällä kohti yhtä isoa kokonaisuutta.

Erityistä huomiota tulee kiinnittää dataset- ja plot-luokkien toimivuuteen. SOMDataset-luokkaa testataan SOMDataItem-luokan kanssa. HCDataset-luokkaa testataan HCTreeNode-, HeatMap- ja DataRange-luokkien kanssa. Tämän jälkeen dataset-luokat testataan plot-luokkien kanssa.

Lisäksi testataan, toimivatko toteutetut luokat JFreeChart-kirjaston osana ja osaavatko ne käyttää JFreeChart-kirjaston toimintoja hyväkseen. JUnit-kirjastoa käytetään integroititestausten toteuttamisessa.

2.3 Järjestelmätestaus

Kaikki vaatimukset, jotka määriteltiin vaatimusmäärittelyvaiheessa, testataan järjestelmätestausvaiheessa. Järjestelmätestaus toteutetaan ns. black box -testauksena, jossa toteutuksen yksityiskohtia ei oteta huomioon vaan testataan ainoastaan valmiin tuotteen toimivuutta vaatimusmäärittelyssä haluttujen ominaisuuksien näkökulmasta. Järjestelmätestaus tehdään tietojenkäsittelytieteen laitoksen järjestelmässä.

Testauksessa kirjataan ylös havaitut puutteet ja poikkeukset verrattuna vaatimusedokumenttiin sekä ilmeiset puutteet ja virheet, vaikka ne täyttäisivät vaatimukset (esim. käyttöliittymässä).

3 Muu testaus

3.1 Volyymitestaus

Visualisointeja testataan normaaleilla sekä ylisuurilla datamäärillä. Normaaleja datamääriä ovat lämpökartassa 200x20 datapistettä ja SOM-kartassa 50x10 datapistettä. Visualisoinnit testataan myös 2- ja 10-kertaisilla datamäärillä. Lisäksi lämpökartta testataan 30000 alkiolla.

4 Testausaikataulu

Testit kirjoitetaan yhtäaikaaisesti koodin kanssa. Maanantaina 14.8. pidetään alustava järjestelmätestaus.