

Convergence of messaging

Testaussuunnitelma

The Converge Group:

Mikko Hiipakka

Anssi Johansson

Joni Karppinen

Olli Pettay

Timo Ranta-Ojala

Tea Silander

Helsinki 20. joulukuuta 2002

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1 Johdanto	1
2 Testausryhmä	1
3 Testattavat ohjelman osat	1
4 Testaamatta jätettävät ohjelman osat	2
5 Testausympäristö	2
6 Testaustapa	2
7 Testausvaiheet	3
7.1 Luokkatestaus	3
7.2 Moduulitestaus	3
7.3 Integraatiotestaus	4
7.4 Järjestelmättestaus	4
8 Tuotettavat dokumentit	4
9 Testauksen aikataulu	5
10 Testauksen riskianalyysi	5
10.1 Työn viivästyminen	5
10.2 Ryhmän jäsenen sairastuminen	6
10.3 Laitteisto- tai järjestelmäongelmat	6

0.0.1	05.12.2002	Dokumentti luotu	Tea Silander
0.0.2	06.12.2002	Dokumentin sisältöä täydennetty	Tea Silander
0.0.3	18.12.2002	Dokumentin sisältöä korjattu	Tea Silander
0.0.4	19.12.2002	Dokumentin sisältöä korjattu	Tea Silander
1.0	19.12.2002	Dokumentti jäädytetty	

1 Johdanto

Tässä dokumentissa kuvataan viestien hallintaan suunnitellun Converge-järjestelmän testaussuunnitelma ja testaukseen liittyvät dokumentit.

Converge-järjestelmä on toteutettu Helsingin Yliopiston Ohjelmistotuotantoprojektina syksyllä 2002. Järjestelmä on tarkoitettu käyttäjän henkilökohtaisten viestien hallintaan. Käyttäjä voi luoda profiileita ja kontekstimalleja, joiden perusteella hänen viestejään käsitellään käyttäjän toivomalla tavalla. Järjestelmän prototyypin on toteutettu WWW-käyttöliittymä, jonka kautta järjestelmää käytetään.

Testaussuunnitelma ja ohjelmiston testaus toteutetaan osana Helsingin Yliopiston Tietojenkäsittelytieteen laitoksen Ohjelmistotuotantoprojektia. Testauksella pyritään varmistumaan siitä, että ohjelmisto toteuttaa määrittelydokumentin mukaisen toiminnallisuuden sekä ohjelmistolle asetetut implisiittiset laatuvaatimukset.

2 Testausryhmä

Testaukseen osallistuu koko projektiryhmä ainakin niiltä osin, että jokainen henkilö testaa itse tuottamansa luokat luokkatestauksella.

3 Testattavat ohjelman osat

Ohjelmisto pyritään testaamaan käyttöliittymätestauksella mahdollisimman kattavasti siltä osin kuin se on mahdollista. Ohjelmiston testauksessa pyritään kiinnittämään erityistä huomiota ryhmän toteuttamien osien testaukseen. Näitä osia ovat tiedonhakumoduuli, ydin sekä asiakasmoduuli. Lisäksi testataan asiakasohjelmiston käyttöliittymä.

4 Testaamatta jätettävät ohjelman osat

Projektiryhmä jättää testaamatta kolmannen osapuolen komponentit muilta osin kuin järjestelmätestauksen kautta. Näitä komponentteja ovat tiedonhakumodulissa JavaMail sekä tähän liittyvä Javabeans Activation Framework, ytimessä Xindice-tietokanta sekä Drools ja asiakasmoduulissa Webcore-palvelinohjelmisto.

5 Testausympäristö

Testauksessa käytetään Linux CSL2-ympäristöä ja Java-ajoympäristön versiota 1.4.1. Järjestelmätestaus suoritetaan käyttäen sekä yhtä, että useampaa tietokonetta yhtä aikaa. Koneissa on oltava WWW-selain ja riittävä suorituskyky. Asiakasohjelmiston testaus suoritetaan Mozilla-selaimella (versio 1.0.1 tai uudempi).

6 Testaustapa

Testauksessa pyritään löytämään virheitä tai puutteita ohjelman toimintalogiikasta tarkasteltuna määrittelydokumenttia vasten. Testauksella pyritään kattamaan mahdollisimman suuri osa ohjelmistosta. Testeissä ei tutkita ohjelman suorituskykyä, virhesietoisuutta tai muita vastaavia ominaisuuksia. Testauksessa ei käytetä apuna valmiita testaustyöhön tarkoitettuja ohjelmistoja.

Testaus suoritetaan ennalta laadittujen testitapausten mukaan. Testitapaukset laaditaan pääosin määrittelydokumentin pohjalta käyttäen myös JavaDocia sekä lähdekoodia. Testitapaukset tehdään ennen testauksen aloittamista. Testitapauksia voidaan myös lisätä kesken testauksen, mikäli tarvetta ilmenee. Yksittäinen testitapaus katsotaan hyväksyttävästi suoritetuksi kun testin suorituksen tulos on sama vastaavan testausdokumenttiin kirjatun testin odotettu tulos.

Löytyneet virheet raportoidaan testausdokumenttiin ja pyritään korjaamaan. Virheiden korjauksen jälkeen ne ohjelman osat, joihin muutokset vaikuttavat pyritään testaamaan uudelleen. Testaus katsotaan päättyneeksi, kun kaikki testitapaukset on käyty läpi, tai viimeistään 19.12.2002.

7 Testausvaiheet

Ohjelmiston testaus suoritetaan vaiheissa. Testaus alkaa yhden luokan toiminnallisuuden testaamisesta ja etenee isompiin kokonaisuuksiin. Lopuksi testataan koko ohjelmiston toiminta järjestelmätestaus-vaiheessa. Converge-järjestelmän testaamisessa moduulitestaus ja integraatiotestaus jäävät vähemmälle huomiolle ajan puutteen vuoksi.

7.1 Luokkatestaus

Luokkatestaus pyritään suorittamaan kullekin luokalle siinä vaiheessa, kun luokkaa toteutetaan. Luokkatestaus suoritetaan white-box -menetelmällä kirjoittamalla luokalle testiohjelma, joka testaa vähintään luokan tärkeimpien metodien toiminnan. Testejä ei tarvitse raportoida, vaan virheet korjataan havaittaessa.

7.2 Moduulitestaus

Moduulitestauksessa testauksen kohteena on yksittäinen moduuli, jonka toimintaa verrataan suunnitteludokumentissa sille asetettuihin vaatimuksiin. Moduulin testaa sen toteuttanut henkilö tai henkilöt. Converge-järjestelmässä testataan seuraavat moduulit: tiedonhakumoduuli, ydin ja asiakasmoduuli, joka pitää sisällään myös asiakasohjelmiston. Moduulitestauksen suorittavat kyseisen moduulin vastuuhenkilöt luokkatestauksen yhteydessä. Moduulitestauksen testitapauksia ja tuloksia ei dokumentoida, vaan virheet korjataan havaittaessa.

7.3 Integraatiotestaus

Integrointitestauksessa yhdistellään yhteen moduuleita tai moduuliryhmiä. Testaamalla pyritään selvittämään erityisesti, että moduulin tarjoama rajapinta toimii, kuten suunnitteludokumentissa on määritelty. Integrointitestaus voidaan tarvittaessa toteuttaa rinnakkain moduulitestauksen kanssa.

Integraatiotestausta ei ajanpuutteen vuoksi toteuteta. Kuitenkin kunkin moduulin testausvaiheessa on testattu myös moduuleiden tarjoamat rajapinnat.

7.4 Järjestelmätestaus

Järjestelmätestauksessa ohjelmistoa tarkastellaan kokonaisuutena, pyrkimyksenä varmistua siitä, että ohjelmisto vastaa määrittelydokumentissa asetettuja vaatimuksia.

Järjestelmätestaus suoritetaan black-box -menetelmällä käyttäen asiakasohjelmiston WWW-käyttöliittymää.

8 Tuotettavat dokumentit

Tämän dokumentin lisäksi tuotetaan testausdokumentti, joka pitää sisällään testitapaukset sekä niiden tulokset ja mahdollisesti löydetyt puutteet. Testausdokumenttiin ei kirjata luokkatestauksen testitapauksia eikä testituloksia.

Testitapaukset kirjataan testausdokumenttiin ennen testauksen aloittamista. Kunkin testitapaukseen kirjataan myös odotettu tulos. Testitapauksien suorittamisen jälkeen testin tulos kirjataan testausdokumenttiin.

Testausdokumentissa testitapaukset on luokiteltu testattavan toiminnon mukaisesti liittämällä järjestelmätestausvaiheen tunnuksen J perään testitapauksen numero. Virheet löytäneet testit suoritetaan virheiden korjauksen jälkeen uudelleen ja dokumentoidaan liittämällä testin alkuperäisen tunnuksen perään tunnus, joka

ilmaisee, monettako kertaa testi suoritetaan, esimerkiksi J20-II.

9 Testauksen aikataulu

Seuraavassa esiteltynä testauksen aikataulu.

Vaihe	Valmis	Vastuhenkilö
Testaussuunnitelma valmis	11.12.2002	Tea
Testitapaukset valmiit	12.12.2002	Tea
Moduulitestaus valmis	12.12.2002	Moduulin kirjoittaja
Järjestelmättestaus valmis	19.12.2002	Ryhmä
Testausdokumentti valmis	20.12.2002	Tea

10 Testauksen riskianalyysi

Ajan puutteen vuoksi on olemassa erityisen suuri riski testauksessa epäonnistumiselle. Tämän vuoksi ryhmä kiinnittää erityisen suurta huomiota testauksen aikatauluun. Tämän lisäksi juuri testauksen kannalta olennaisia riskejä on pyritty listaamaan alle.

10.1 Työn viivästyminen

Riskinä aikataulutuksesta huolimatta työn viivästyminen.

Riskin toteutuminen olisi projektin valmistumisen kannalta kriittinen. Riskiä pyritään vähentämään jakamalla työtä sopivasti. Tarvittaessa testitapauksille määritellään prioriteetit ja mietitään, mitkä testitapaukset voitaisiin jättää suorittamatta.

10.2 Ryhmän jäsenen sairastuminen

Riskinä ryhmän jäsenten sairastumiset, loukkaantumiset tai muut syyt, jotka estävät henkilön osallistumisen työhön, joka taas saattaa aiheuttaa muille enemmän työtä ja sitä kautta johtaa aikataulun viivästymiseen.

Ryhmän jäsenet eivät pysty hallitsemaan riskiä. Riskin toteutuessa estynyt henkilö pyrkii mahdollisuuksien mukaan jatkamaan työskentelyään kotoa käsin.

10.3 Laitteisto- tai järjestelmäongelmat

Riskinä laitteistojen tai käyttöjärjestelmäympäristöjen toimimattomuus tai Tietojenkäsittelytieteen laitokselle pääsyn estyminen.

Riskiä toteutuessa jatketaan työtä toisella tietokoneella tai tarvittaessa toisena ajan-kohtana, kuitenkin niin, että jokainen saa suoritettua oman osansa työstä. Myös mahdollisia järjestelmän käyttökatkoja pyritään ennakoimaan seuraamalla laitoksen uutisia.