

Lohtu-projekti

Testaussuunnitelma

Versiohistoria:

1.0	19.2.2003	1. versio	Mari
1.1	20.2.2003	Muutoksia	Mari
1.2	25.2.2003	Katselmoinnissa esiin tulleet muutokset	Mari
1.3	17.3.2003	2. syklissä tehtävät testit	Virve
1.4	19.4.2003	Lisätty 2. syklin testeihin menun muutosten tarkistaminen sekä niiden testaus 2. syklissä.	Virve
1.5	6.5.2003	Poistettu 2. syklin testeistä syklin lisäys ja poisto	Virve

Helsinki 6. toukokuuta 2003

Kimmo Airamaa, Andreas Asuja, Mari Muuronen, Seppo Pastila, Virve Taivaljärvi

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Laudatur-projektityö, kevät 2003

Sisältö

1 JOHDANTO	1
2 TESTATTAVAT OHJELMAN OSAT	1
3 TESTAAMATTA JÄTETTÄVÄT OHJELMAN OSAT	1
4 TESTAUSTAPA	2
4.1 TARKASTUS.....	2
4.2 BLACK-BOX TESTAUS	2
4.3 WHITE-BOX TESTAUS	3
4.4 TESTAUKSEN PÄÄTTYMINEN	3
4.5 TESTIYMPÄRISTÖ	3
5 TESTAUSVAIHEET	4
6 BLACK-BOX TESTIAINEISTO	5

1 Johdanto

Testattavana on projektihallinnan työkalu, jonka avulla voidaan suunnitella projekteja, seurata niiden etenemistä, seurata tehtyjen suunnitelmien toteutumista sekä verrata eri projektiryhmiä toisiinsa.

Sovellus on toteutettu J2EE-tekniikalla, Java Servlet 2.3 ja JSP 1.2 versioilla. Käytetty J2SE versio on 1.4.

Sovelluksesta tuotetut dokumentit löytyvät projektiryhmän kotihakemistosta dokumentit-hakemistosta.

Testauksen ja siihen liittyvät toimenpiteet suorittaa Lohtu-projektiryhmä.

2 Testattavat ohjelman osat

Ohjelmisto testataan sykleittäin, siten että syklin lopussa testataan syklissä tuotetut ominaisuudet ja toiminnallisuudet sekä testataan syklissä toteutetun ohjelmiston osan toiminta aikaisemmin toteutettujen osien kanssa.

3 Testaamatta jätettävät ohjelman osat

Tarkoituksena on testata sovellus määrittelydokumenttiin pohjautuen, niin ettei mikään osa-alue jää testauksen ulkopuolelle.

4 Testaustapa

Testauksen tarkoituksena on testata kaikki ohjelman niin kattavasti kuin mahdollista. Testaukseen kuuluvat osat ovat testaussuunnitelma, testaussuunnitelman katselmointi, black-box testit ja white-box testit. Testit suoritetaan pääosin järjestelmätestauksen tasolla. Yksikkötestit suoritetaan ohjelmoinnin yhteydessä, ja niiden suorittaminen on ohjelmoijan vastuulla. Sama pätee luokkien integrointitestaukseen. Luokista arviolta noin neljännes on yksikkötestattavissa JUnit-työkalulla. Muut luokat jätetään testaamatta.

4.1 Tarkastus

Tarkastettavana on testaussuunnitelma. Tarkastuksen tavoitteena on löytää suunnitelman mahdolliset puutteet ja/tai virheet. Tarkastuksessa havaitut virheet arvioidaan niiden vakavuuden perusteella. Mikäli suunnitelma hyväksytään sellaisenaan, ei siitä aiheudu lisätoimenpiteitä. Mikäli virheitä ja/tai puutteita esiintyy korjataan testaussuunnitelmaa siten, että se läpäisee tarkastuksen. Ja mikäli testaussuunnitelmaa ei tarkastuksessa hyväksytä, on seurauksena suunnitelman korjaus ja uusi tarkastus.

4.2 Black-box testaus

Black-box testaus tehdään ohjelmiston web-käyttöliittymälle. Testitapausten luonnissa käytetään ekvivalenssiluokkajakoa, siten että samanlaisen toiminnan aiheuttava testiaineisto kuuluu samaan ekvivalenssiluokkaan. Sovelluksen testaukseen tarkoitettu testiaineisto löytyy luvusta 6.

4.3 White-box testaus

White-box testaus jätetään ohjelmoijan vastuulle. Periaatteena käytetään, että CVS:ään lisättävät luokat testataan mikäli mahdollista, ja ainakin varmistetaan että CVS:ään lisättävät luokat kääntyvät.

4.4 Testauksen päätyminen

Testauksen katsotaan päättyneeksi kun kuhunkin testausvaiheeseen tehdyt testit ja testiaineistot on ajettu läpi onnistuneesti. Vaihtoehtoinen tapa testauksen päättymiseen on kurssin suorittamiseen varatun rajallisen ajan päätyminen, jolloin myös testaus keskeytetään.

Testauksen päätyminen merkitsee, että projektiryhmä hyväksyy tuotteensa, ja tuote on valmis luovutettavaksi ja tuotantokäyttöön.

4.5 Testiympäristö

Testausympäristönä on TKTL:n Linux-palvelimet. Sovellus testataan TKTL:n koneilla olevilla selaimilla, Microsoft Internet Explorer 6.0 ja Mozilla 1.0.

Testauksen suorittamiseksi tarvitaan testattavan sovelluksen lisäksi Tomcat- palvelin ja tietokanta, josta löytyy sovelluksen tarvitsemat taulut, ks. suunnitteludokumentti. Lisäksi suoritettava ohjelma tarvitsee oikein asetetut asetustiedostot. Tarkemmat ohjeet käyttöohjeesta.

5 Testausvaiheet

Testaussuunnitelma: Tehdään ensimmäisenä.

Tarkastus: Tarkastus tehdään testaussuunnitelmalle. Kun testaussuunnitelma on tarkastettu hyväksytysti on tarkastusvaihe suoritettu onnistuneesti.

White-box testaus: Mahdolliset testitapaukset tehdään koodauksen yhteydessä. Testit suoritetaan heti moduulin valmistuttua.

Black-box testaus: Testitapaukset tehdään testaussuunnitelmaan (luku 6).

6 Black-box testiaineisto

Testeissä oletetaan, että sovellus ja sen tarvitsemat muut sovellukset ja palvelimet toimivat oletetusti.

Testattava ominaisuus	Testiproseduuri	Odotettu tulos
<i>Sovellukseen kirjautuminen</i>		
Testataan 1.syklissä	Kirjaudutaan sovellukseen Ohturyhmäläisen käyttäjätunnuksella ja salasanalla.	Kirjautuminen onnistuu ja kurssilainen pääsee oman kurssinsa tietoihin.
Testataan 1.syklissä	Kirjaudutaan sovellukseen Ohtuvastaavan käyttäjätunnuksella ja salasanalla.	Kirjautuminen onnistuu ja kurssivastaava pääsee sovelluksen hallintatietoihin.
Testataan 1.syklissä	Kirjaudutaan sovellukseen virheellisellä käyttäjätunnuksella.	Kirjautuminen epäonnistuu.
Testataan 1.syklissä	Kirjaudutaan sovellukseen oikealla käyttäjätunnuksella, mutta väärällä salasanalla.	Kirjautuminen epäonnistuu.
Testataan 1.syklissä	Yritetään kirjautua opiskelijan tunnuksilla sovelluksen hallintoon.	Kirjautuminen epäonnistuu.
	Salasana on unohtunut, pyydetään järjestelmää lähettämään se.	Sovellus avaa ikkunan, jossa kysytään käyttäjätunnusta. Salasana saapuu kyselijän sähköpostiin.

<i>Projektin ja käyttäjien lisäys</i>		
	Tietojen haku kurssijärjestelmästä-sivulla haetaan oikeasyntaksinen xml-tiedosto, jonka tiedot tallennetaan sovellukseen.	Projekti ja siihen liittyvät käyttäjät lisätään onnistuneesti sovellukseen.
	Tietojen haku kurssijärjestelmästä-sivulla haetaan vääräsyntaksinen xml-tiedosto, jonka tiedot tallennetaan sovellukseen.	Projektia ja käyttäjiä ei voida lisätä. Käyttäjälle annetaan virheilmoitus viallisesta tiedostosta.
	Tietojen haku kurssijärjestelmästä-sivulla yritetään tallentaa olemattoman tiedoston tietoja sovellukseen.	Projektia ja käyttäjiä ei voida lisätä. Käyttäjälle annetaan virheilmoitus tiedoston puuttumisesta.
	Tietojen haku kurssijärjestelmästä-sivulla yritetään tallentaa ei xml-tiedoston tietoja sovellukseen.	Projektia ja käyttäjiä ei voida lisätä. Käyttäjälle annetaan virheilmoitus viallisesta tiedostosta.
<i>Projektin suunnittelu</i>		
Testataan 1.syklissä	Täytetään projektin tiedot perustiedot-sivulle ja tallennetaan muutokset	Mikäli jokin pakollinen tieto on jäänyt täyttämättä, sovellus ilmoittaa siitä käyttäjälle. Tiedot tallentuvat oikein ja näkyvät oikein perustiedot-sivulla.

- <i>prosessimalli</i>		
Testataan 1.syklissä	Valitaan haluttu prosessimalli projektin perustiedot-sivulla ja tallennetaan muutokset.	Prosessimalli tallentuu oikein tietokantaan. Tieto näkyy oikein projektin perustiedot-sivulla.
Testataan 1.syklissä	Vaihdetaan prosessimallia projektin perustiedot-sivulla ja tallennetaan tiedot.	Tieto tallentuu oikein tietokantaan. Uusi prosessimalli näkyy projektin perustiedot-sivulla.
- <i>vaihejako</i>		
Testataan 2.syklissä	Tehdään uusi vaihe 1.sykliin vaihejako-sivulla ja tallennetaan muutokset.	Mikäli jokin pakollinen tieto jää täyttämättä ohjelma ilmoittaa siitä käyttäjälle. Vaihe tallentuu oikein tietokantaan ja täytetyt tiedot näkyvät oikein vaihejaot-sivulla.
Testataan 2.syklissä	Poistetaan vaihe 1.syklistä ja tallennetaan muutokset.	Ohjelma varmistaa poiston erillisellä dialogilla. Vaihe ja siihen liittyvät tiedot poistuvat 1. syklin tiedoista, tietokannasta sekä vaihejako-sivulta.
Testataan 2.syklissä	Muutetaan vaiheen tietoja ja tallennetaan muutokset.	Muutetun vaiheen tiedot päivittyvät tietokantaan ja vaihejako-sivulle.
Testataan 2.syklissä	Lisätään uusi vaihe Lisää vaihe-painikkeella vaihejako-sivulla.	Uusi tyhjä vaihe, jossa on neljä tyhjää tehtäväriiviä ilmestyy vaihejako-sivulle.

- ohjelman ositus		
Testataan 1.syklissä	Ositetaan ohjelma perustiedot- sivulla ja tallennetaan muutokset.	Mikäli jokin pakollinen tieto jää täyttämättä ohjelma ilmoittaa siitä käyttäjälle. Ohjelman osat tallentuvat oikein tietokantaan ja tiedot näkyvät oikein perustiedot-sivulla (ja vaihejako-sivulla tehtävien kohdalla Osa-sarakkeessa, ei 1.syklissä).
Testataan 1.syklissä	Lisätään uusi osa perustiedot- sivulla ja tallennetaan muutokset.	Uusi osa tallentuu oikein tietokantaan ja lisätty osa näkyy perustiedot-sivulla (ja vaihejako-sivulla tehtävien kohdalla Osa-sarakkeessa, ei 1.syklissä).
Testataan 1.syklissä	Poistetaan osa perustiedot-sivulla ja tallennetaan muutokset.	Mikäli osa on liitetty johonkin tehtävään, annetaan käyttäjälle ilmoitus, että kyseistä osaa ei voida poistaa ennen kuin siihen kohdistuvat viitteet on poistettu. Mikäli viitteitä ei ole, osa poistetaan tietokannasta ja se poistuu perustiedot-sivulta (ja vaihejako-sivulta, ei 1.syklissä).

- tehtävät		
Testataan 2.syklissä	Luodaan tehtävä vaihejako-sivulla ja tallennetaan muutokset.	Mikäli jokin pakollinen tieto jää täyttämättä ohjelma ilmoittaa siitä käyttäjälle. Tehtävä tietoineen tallentuu oikein tietokantaan. Lisätyt tiedot näkyvät oikein vaihejako-sivulla.
Testataan 2.syklissä	Muutetaan olemassa olevan tehtävän tietoja vaihejako-sivulla ja tallennetaan muutokset.	Muutokset tallentuvat oikein tietokantaan. Muutetut tiedot näkyvät oikein vaihejako-sivulla.
Testataan 2.syklissä	Poistetaan tehtävä vaihejako-sivulla ja tallennetaan muutokset.	Tehtävä poistetaan tietokannasta ja se poistuu myös ohjelman vaihejako-sivulta.
Testataan 2.syklissä	Lisätään uusi tehtäväriivi Lisää tehtävä-painikkeella vaihejako-sivulla.	Uusi tyhjä tehtäväriivi ilmestyy vaihejako-sivulle vaiheen jo olemassa olevien tehtäväriivien alle.
- aikataulut		
Testataan 2.syklissä	Lisätään vaihejako-sivulla vaiheen alku- ja loppupäivämäärät siten, että alkupäivämäärä on ennen loppumispäivämäärää ja tallennetaan muutokset	Tiedot tallennetaan oikein ja tiedot näkyvät vaihejako-sivulla.
Testataan 2.syklissä	Lisätään vaihejako-sivulla vaiheen alku- ja loppupäivämäärät siten, että alkupäivämäärä on loppumispäivämäärään jälkeen ja tallennetaan muutokset	Tietoja ei tallenneta ja käyttäjälle annetaan ilmoitus virheellisistä päivämääristä.

Testataan 2.syklissä	Lisätään vaihejako-sivulla tehtävän deadline siten, että se on vaiheen alkamis- ja loppumispäivämäärän välissä tai jompikumpi näistä päivämääristä ja tallennetaan muutokset	Tiedot tallennetaan oikein ja tiedot näkyvät vaihejako-sivulla.
Testataan 2.syklissä	Lisätään vaihejako-sivulla tehtävän deadline siten, että se on ennen vaiheen alkamispäivää tai loppumispäivämäärän jälkeen ja tallennetaan muutokset	Tietoja ei tallenneta ja käyttäjälle annetaan ilmoitus virheellisistä päivämääristä
Testataan 2.syklissä	Vaihejako-sivulla syötetään päivämäärä väärässä muodossa johonkin päivämääräkentistä	Käyttäjälle annetaan ilmoitus virheellisestä päivämäärän muodosta
- sykli		
	Poistetaan sykli Poista sykli-painikkeella vaihejako-sivulla.	Ohjelma varmistaa poiston erillisellä dialogilla. Sykli ja kaikki siihen liittyvät tiedot poistetaan tietokannasta ja ne poistuvat myös vaihejako-sivulta. Samalla linkki syklin vaihejakosivulle poistuu menusta.
	Lisätään uusi sykli sivujen vasemmalla reunalla olevan menun Lisää sykli-painikkeesta.	Uusi sykli linkki ilmestyy sivujen vasemman laidan menuun. Tästä linkistä avautuu syklin vaihejakosivu.