

## **Testaussuunnitelma**

Ohjelmistotuotantoprojektin tietojärjestelmä — OhtuTie

Helsinki 14.7.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (6 ov)

**Projektiryhmä**

Janne Nevalainen  
Jyrki Kankaanpää  
Sinikka Loikkanen  
Esa-Matti Miettinen  
Petteri Nurmi  
Max Österman

**Asiakas**

Turjo Tuohiniemi

**Vastuhenkilö**

Juha Taina

**Kotisivu**

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

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
0.5	1.7.2004	Työstöversio

# Sisältö

<b>1 Johdanto</b>	<b>1</b>
1.1 Erikoissanasto . . . . .	1
1.2 Dokumentin rakenne . . . . .	1
<b>2 Testausstrategia</b>	<b>2</b>
<b>3 Testauksen vaiheet</b>	<b>2</b>
3.1 Yksikkötestaus . . . . .	2
3.2 Integrointitestaus . . . . .	3
3.3 Järjestelmätestaus . . . . .	3
<b>4 Testaamatta jäävä toiminnallisuus</b>	<b>3</b>
<b>5 Testausympäristö</b>	<b>3</b>
<b>6 Järjestelmätestauksen testitapauksia</b>	<b>4</b>
6.1 Yhteisten käyttötapauksen testaaminen . . . . .	4
6.2 Opiskelijan käyttötapauksen testaaminen . . . . .	5
6.3 Ohjaajan käyttötapauksen testaaminen . . . . .	7
6.4 Vastuuhenkilön käyttötapauksen testaaminen . . . . .	8

# 1 Johdanto

Tässä dokumentissa esitetään OhtuTie-projektin testaussuunnitelma. Projekti on osa Helsingin yliopiston Tietojenkäsittelytieteen laitoksen kurssia 581260-4 Ohjelmistotuotantoprojekti. Projektissa tuotetaan tietojärjestelmä Ohjelmistotuotantoprojektia suorittavien opiskelijoiden ja kurssin ohjaajien ja vastuuhenkilöiden käyttöön.

Testauksen tavoitteena on taata ohjelmiston virheettömyys ja ohjelmistolle esitettyjen vaatimusten toteutuminen. Kaikkien ohjelmiston tarjoamien palvelujen tulisi siis toimia dokumentoidulla tavalla eikä ohjelmistosta myöskään saisi puuttua mitään määrittelydokumentissa esitettyä korkeimman prioriteetin toimintoa. Testauksessa ei käytännössä voida koskaan havaita kaikkia ohjelmiston virheitä, mutta pyrimme tuottamaan ainakin keskeisimmiltä osiltaan toimivan ohjelmistotuotteen. Siksi testaus keskittyy erityisesti ohjelmiston yleisimmin käytettäviin osiin.

## 1.1 Erikoissanasto

**Metriikka:** Mitattava tieto. Metriikalla on nimi, kuvaus ja tyyppi eli metriikkamalli. Kuvaus on samalla ohje käyttäjälle. Esimerkiksi valmiista koodista laskettu koodirivien lukumäärä on metriikka.

**Metriikkasarja:** Kokoelma toisiinsa liittyviä metriikoita. Esimerkkinä käyvät toteutusvaiheessa kerättävät tiedot tai tarkastuksesta kerättävät tiedot.

**Projekti:** Helsingin yliopiston tietojenkäsittelytieteen laitoksen kurssin Ohjelmistotuotantoprojekti puitteissa toteutettava ryhmätyö, jonka tavoitteena on ohjelmistosovelluksen toteuttaminen.

**Tunnistenumero:** Vastuuhenkilön antama projektit lukukauden sisällä yksilöivä numero.

**Regressiotestaus:** Ohjelmistoon tehtyjen muutosten aiheuttamien ongelmien havaitsemiseen keskittyvä testaus.

## 1.2 Dokumentin rakenne

Toisessa luvussa kuvataan testaussuunnitelman päälinjat ja raportointimenettely. Luku kolme käsittää tarkemman kuvauksen testauksen osavaiheista. Luvussa neljä käsitellään rajallisesta testaukseen käytettävissä olevasta ajasta tai muista syistä testaamatta jääviä asioita. Viidennessä luvussa määritellään ympäristöt, joissa järjestelmän toimivuutta testataan. Kuudes luku sisältää listan järjestelmätestauksessa suoritettavista testeistä.

## 2 Testausstrategia

Testaus jakautuu kolmeen vaiheeseen, yksikkötestaukseen, integraatiotestaukseen ja järjestelmätestaukseen. Yksikkötestauksessa testataan yksittäisten luokkien toimintaa erillään muista luokista, ja integraatiotestauksessa yksikkötestattujen luokkien yhteistoimintaa. Järjestelmätestauksessa testattavana on koko ohjelmisto. Järjestelmätestauksessa tuotettua ohjelmistoa testataan testiaineistolla kehitysryhmän toimesta.

Testauksessa käytetään apuna JUnit-testausohjelmistoa, joka tarjoaa tavan automatisoida testausta. JUnitia käytettäessä testejä on tarkoitus suunnitella ja toteuttaa samalla kun itse ohjelmistoa ohjelmoidaan. Siksi testaaminen käytännössä alkaa jo toteutusvaiheessa. Erityistä huomiota tulisi kiinnittää testattavien komponenttien hyväksymien arvoalueiden rajoilla oleviin syötteisiin sekä merkkijonojen tapauksessa tyhjiin syötteisiin.

Testauksen tuloksista raportoidaan testausdokumentissa. Yksittäisistä testeistä kerrotaan testaajan nimi, suorituspäivämäärä, testattava asia, kuvaus testistä, odotettu tulos sekä havaitut virheet. Testauksessa löydetyt virheet pyritään korjaamaan mahdollisuuksien mukaan. Mikäli jokin vika havaitaan vasta testauksen loppuvaiheessa ja sen korjaaminen olisi mahdotonta käytettävissä olevan ajan puitteissa, virhe jätetään korjaamatta ja dokumentoidaan.

## 3 Testauksen vaiheet

Testaus koostuu kolmesta osavaiheesta, yksikkö-, integrointi ja järjestelmätestauksesta. Vaiheet seuraavat ajallisesti toisiaan esitetyssä järjestyksessä.

### 3.1 Yksikkötestaus

Yksikkötestauksessa keskitytään yhden ohjelman yksikön toiminnan tarkasteluun. Vaiheessa varmistetaan, että yksikkö toimii virheettömästi ja suunnitteludokumentissa kuvautulla tavalla. Yksikkötestaus suoritetaan osin JUnit-työkalun tukemana automatisoituina yksikkö- ja regressiotestauksena.

Jokainen ohjelmoija suorittaa yksikkötestauksen itse ohjelmoimalleen komponentille. Yksikkötestauksessa käytetään yleisesti lasilaatikkotestauksen (white-box) menetelmiä eli testauksessa käytetään hyväksi yksikön lähdekoodia. Lähdekoodin avulla suunnitelluilla testitapauksilla pyritään paikallistamaan mahdolliset virheet yksikön sisällä. Yksikkötestauksessa käytetään ajan salliessa myös niin sanottuja tynkämoduuleja eli tyhjiä luokkia, joiden tehtävänä on simuloida muita testauksessa mahdollisesti tarvittavia moduuleja.

Testit pyritään kirjoittamaan etukäteen, jolloin toteutettavan yksikön julkinen liittymä tulee suunniteltua loogiseksi ja suppeaksi. Tällöin valmiiksi kirjoitettu testiskripti määrittelee milloin yksikkö on valmis. Testejä ajetaan ohjelmoinnin kanssa rinnakkain, jolloin aina jonkin toiminnallisuuden toteutuksen jälkeen sen toiminta testataan. Jokaisen yksikön toiminta pyritään erottamaan muusta koodista ja ympäristöstä niin, että testin läpime-

no ei riipu muusta koodista. Tällöin säästetään aikaa testiä kirjoitettaessa ja nopeutetaan mahdollisten virheiden löytämistä.

## 3.2 Integroititestausta

Integroititestauksessa keskitytään komponenttien välisiin rajapintoihin. Tässä vaiheessa testataan erillisten komponenttien välistä vuorovaikutusta sekä kommunikointia ja toiminnallisuutta niiden välillä. Integroitavien yksiköiden yksikkötestaus täytyy olla hyväksyttyvästi tehtynä ennen tämän vaiheen aloittamista. Integroititestausta suoritetaan kokoavana testauksena (bottom-up), jolloin alemman tason komponentit kehitetään ja testataan ennen siirtymistä korkeamman tason rakenteisiin. Testaus aloitetaan integroimalla alimman tason yksiköt yksitellen mukaan ohjelmistoon ja testaamalla nämä. Tämän jälkeen integroidaan seuraavalla tasolla olevat yksiköt alimman tason kanssa. Tätä kiertoa jatketaan kunnes kaikkein ylin taso on integroitu ja testattu. Rajapintoja testataan ajurien avulla, jotka simuloivat korkeamman tason komponenttien toimintaa. Testauksen tässä vaiheessa pyritään tarkestelemaan erityisesti rajapintafunktioille annettavat syötteet ja niiden palautusarvot, sekä rajapinnan yhdistämien moduuleiden käyttäytyminen ja kommunikaatio toisesta osapuolesta johtuvissa virhetilanteissa.

## 3.3 Järjestelmätestaus

Järjestelmätestauksessa pyritään testaamaan ohjelmistoa sille määriteltyjen vaatimusten suhteen. Testiaineistona käytetään Ohtutie - projektin omasta projektista sekä aiemmista ohjelmistotuotantoprojekteista syötettävää mittausaineistoa. Testaus pyritään suorittamaan niin, että projektiryhmän jäsenet testaavat eri alueita kuin ovat itse toteuttaneet.

## 4 Testaamatta jäävä toiminnallisuus

Järjestelmän tietoturvaa ja kuormituksen vaikutuksia ei tulla erikseen testaamaan. Järjestelmää tullaan käyttämään niin, että vain muutama henkilö on yhtäaikaaisesti kirjautuneena sisään, eikä näinollen jouduta kovin suuriin kuormitusasteisiin.

## 5 Testausympäristö

Järjestelmän toimintaa testataan määrittelydokumentissa esitettyjen toimintaympäristöstä esitettyjen vaatimusten mukaisesti. Järjestelmän luonteesta johtuen käyttöjärjestelmällä ei pitäisi olla sanottavaa vaikutusta toimintaan. Siksi testaus suoritetaan ensisijaisesti Windows -käyttöjärjestelmässä. Järjestelmätestauksessa pyritään testaamaan toimintaa Internet Explorer ja Mozilla -selaimilla. Tavoitteena on kuitenkin erityisesti osoittaa järjestelmän toimivan määrittelydokumentissas esitettyjen standardien mukaisesti. Järjestelmän

toimintaa mahdollisesti standardeista poikkeavilla selaimilla ei pyritä kattavasti varmistamaan.

Ainoa tietokanta, johon testauksessa keskitytään on Oracle. Tietokannan toimintaa samoin kuin järjestelmän yhteensopivuutta selainten kanssa selvitetään lähinnä järjestelmätestauksessa. Lisäksi jo toteutusvaiheessa pyritään kuitenkin lyhyesti kokeilemaan keskeisimpien tietokantakyselyiden toimintaa.

## 6 Järjestelmätestauksen testitapauksia

Tässä kuvataan ohjelmistolle suoritettavat testitapaukset. Testitapaukset on jaoteltu määrittelydokumentissa esitettyjen käyttötapausten mukaisesti. Näin varmistetaan, että jokainen käyttötapaus tulee ainakin jossakin muodossa testatuksi. Testien tuloksista raportoidaan testausdokumentissa luvussa 2 Testausstrategia kuvatulla tavalla.

Ennen testauksen aloittamista on luotava tarvittavat oikeudet omaavat käyttäjätunnukset. Opiskelijan käyttötapaukset testataan ensisijaisesti käyttäjätunnuksella joka on merkitty opiskelijan tunnuksiksi sekä varustettu opiskelijan oletuskäyttöoikeuksilla. Vastaavasti ohjaajan ja vastuuhenkilön käyttötapaukset testataan ohjaajan ja vastuuhenkilön tunnuksilla. Yhteiset käyttötapaukset testataan kullakin edellä mainitulla tunnustyyppillä. Lisäksi suoritetaan pistokokeen omaisesti testausta muunlaisilla käyttöoikeuksilla varustetuilla tunnuksilla.

Järjestelmään kirjautumista ja uloskirjautumista lukuun ottamatta toimintojen suorittaminen edellyttää oikean välilehden valintaa. Järjestelmään kirjautuminen on myös esiehtona kaikille muille käyttötapauksille. Vanhojen tietojen muokkaamiseen liittyvien käyttötapauksen testaaminen edellyttää muutettavan testiaineiston syöttämistä etukäteen.

### 6.1 Yhteisten käyttötapauksen testaaminen

Yhteisiin käyttötapauksiin liittyvät toiminnot ovat lähtökohtaisesti kaikkien järjestelmän aktiivisten käyttäjien käytettävissä. Toimintoja testataan opiskelijan, ohjaajan ja vastuuhenkilön oletuskäyttöoikeuksilla varustetuilla tunnuksilla.

#### **YH1 Kirjautuminen järjestelmään**

Kirjautuminen järjestelmään edellyttää ainoastaan järjestelmän WWW-sivulle siirtymistä. Kirjautunut käyttäjä ei myöskään voi kirjautua uudelleen ellei hän ensin kirjaudu ulos.

- Kirjaututaan kelvallisilla syötteillä
- Kirjaututaan virheellisillä syötteillä
  - väärä tunnus
  - väärä salasana

- passiivisen henkilön tunnus ja salasana

## **YH2 Kirjautuminen ulos järjestelmästä**

- Yritetään hakea järjestelmästä tietoa uloskirjautumisen jälkeen

## **YH3 Tietojen haku**

Ennen testiä jokaiseen tietokantatauluun on tallennettava ainakin yksi rivi. Tietojen hakuun pääsee valitsemalla välilehden XXX.

- Haetaan yhdellä kyselyllä kaikki saatavissa olevat tiedot
- Suoritetaan haku määrittämättä lainkaan hakuehtoja
- Suoritetaan haku käyttämällä kaikkia tarjottuja hakuehtoja
- Suoritetaan haku käyttäen yhtä mielivaltaisesti valittua hakuehtoa

## **6.2 Opiskelijan käyttötapausten testaaminen**

Opiskelija voi lähtökohtaisesti syöttää järjestelmään omaan projektiinsa liittyviä tietoja sekä muuttaa omia tietojaan. Opiskelija saa myös muuttaa aiemmin syöttämiään tietoja.

### **OP1 Mittaustietojen syöttäminen**

Mittaustietojen syöttämiseksi käyttäjän on valittava välilehti XXX, projekti, johon tiedot liittyvät sekä metriikkasarja. Valintojen jälkeen uudet tiedot syötetään mahdollisilla vanhoilla tiedoilla esitäytetyllä lomakkeella. Testauksessa täytetään tarvittava määrä lomakkeita, jotta kaikki alla luetellut tapaukset saadaan käytyä läpi. Testaus edellyttää projektin ja opiskelijatunnuksen luomista sekä metriikkasarjan liittämistä projektiin.

- Syötetään kelvollisia ja virheellisiä arvoja
  - Numeraalisissa syötteissä
    - \* Suurin sallittu arvo
    - \* Pienin sallittu arvo
    - \* Suurinta sallittua arvoa yhtä suurempi arvo
    - \* Pienintä sallittua arvoa yhtä pienempi arvo
    - \* Merkkijono
  - Merkkijonomuotoisissa syötteissä
    - \* Kelvollinen syöte
    - \* Tyhjä syöte
    - \* Liian pitkä syöte



## **OP2 Tuntikirjanpidon syöttäminen suoraan järjestelmään**

Tuntikirjanpidon syöttämiseksi käyttäjän on siirryttävä välilehdelle XXX. Tiedot syötetään tuntikirjanpitolomakkeelle.

- Syötetään kelvollisia ja virheellisiä arvoja
  - Numeraalisissa syötteissä
    - \* Suurin sallittu arvo
    - \* Pienin sallittu arvo
    - \* Suurinta sallittua arvoa yhtä suurempi arvo
    - \* Pienintä sallittua arvoa yhtä pienempi arvo
    - \* Merkkijono
  - Merkkijonomuotoisissa syötteissä
    - \* Kelvollinen syöte
    - \* Tyhjä syöte
    - \* Liian pitkä syöte

## **OP3 Tuntikirjanpidon syöttäminen tiedostosta**

Ennakkoehdot tuntikirjanpidon syöttämiselle tiedostosta ovat samat kuin vastaavien tietojen antamisella käsin.

- Syötetään kelvollinen tuntikirjanpilotiedosto
- Syötetään virheellinen tuntikirjanpilotiedosto

## **OP4 Henkilötietojen muuttaminen**

Omien henkilötietojen muuttaminen tapahtuu välilehdellä XXX.

- Syötetään kelvolliset arvot
- Syötetään virheellisiä arvoja
  - Liian lyhyt salasana
  - Liian pitkä salasana
  - Sähköpostiosoite, josta puuttuu @-merkki
  - Tyhjät syötteet

## **OP5 Projektin tietojen muuttaminen**

Projektin tietojen muuttaminen edellyttää välilehdelle XXX siirtymistä. Vanhojen tietojen muuttamiseksi on valittava muutettava projekti ja korvattava muokkauslomakkeella olevat tiedot uusilla. Vanhojen tietojen muokkaus vaatii myös ainakin yhden projektin luomista ensin.

- Syötetään omalle projektille kelvollinen nimi
- Syötetään omalle projektille tyhjä nimi
- Syötetään omalle projektille toisen projektin nimi
- Lisätään ja poistetaan omasta projektista metriikkasarja
- Lisätään omaan projektiin sama metriikkasarja useamman kerran

### **6.3 Ohjaajan käyttötapauksen testaaminen**

Ohjaajalla on oletusarvoisesti käytössään kaikki opiskelijan toiminnot, sekä lisäksi projektien hallintaan liittyviä toimintoja.

#### **OH1 Opiskelijan lisäys ja opiskelijatietojen muokkaaminen**

Opiskelijan tietojen muuttaminen edellyttää välilehdelle XXX siirtymistä ja opiskelijan valintaa listasta. Uuden opiskelijan voi lisätä painamalla lisää-nappia.

- Syötetään opiskelijalle kelvollinen nimi, käyttäjätunnus ja salasana
- Syötetään virheellisiä tietoja
  - Syötetään tyhjä nimi
  - Syötetään liian pitkä nimi
  - Syötetään liian lyhyt käyttäjätunnus
  - Syötetään liian pitkä käyttäjätunnus
  - Syötetään liian lyhyt salasana
  - Syötetään liian pitkä salasana

#### **OH2 Projektin tietojen muuttaminen**

Projektin tietojen muuttamiseksi käyttäjän on siirryttävä välilehdelle XXX ja valittava muokattava projekti listasta. Muokkaus tapahtuu korvaamalla vanhoilla tiedoilla esitetyt lomakkeen kenttien arvot uusilla. Uuden projektin luonti puolestaan onnistuu täyttämällä lisää-napin painamisen jälkeen esiin tuleva lomake.

- Lisätään projektiin opiskelija
- Poistetaan projektista opiskelija

#### **OH3 Asiakkaan arvostelun syöttäminen**

Asiakkaan arvostelun syöttämiseksi on siirryttävä välilehdelle XXX ja valittava arvostelun kohteena oleva projekti listasta. Tiedot annetaan täyttämällä arvostelulomake ja painamalla lähetyšnappia.

- Syötetään kelvollisia ja virheellisiä arvoja
  - Numeraalisissa syötteissä
    - \* Suurin sallittu arvo
    - \* Pienin sallittu arvo
    - \* Suurinta sallittua arvoa yhtä suurempi arvo
    - \* Pienintä sallittua arvoa yhtä pienempi arvo
    - \* Merkkijono
  - Merkkijonomuotoisissa syötteissä
    - \* Kevollinen syöte
    - \* Tyhjä syöte
    - \* Liian pitkä syöte

## 6.4 Vastuuhenkilön käyttötapausten testaaminen

Vastuuhenkilöllä on lähtökohtaisesti käytössään kaikki järjestelmän toiminnot. Ohjaajan ja opiskelijan käyttötapausten lisäksi vastuuhenkilö voi muokata metriikoiden ja metriikkasarjojen tietoja, muuttaa kaikkia projekteihin liittyviä tietoja sekä hakea järjestelmästä tietokannan tiedot sisältävän tiedoston.

### VH1 Projektin tietojen muuttaminen

Välilehdellä XXX voidaan muuttaa projektin tietoja projektin valitsemisen jälkeen.

- Syötetään kevollinen tunnistenumero
- Syötetään virheellinen tunnistenumero
  - Käytössäoleva projektin tunnistenumero
  - Tunnistenumero nolla
  - Negatiivinen tunnistenumero
- Poistetaan projektista opiskelija

### VH2 Metriikkatietojen muuttaminen

Metriikkatietojen muuttaminen tapahtuu välilehdellä XXX ja edellyttää muokattavan metriikan valintaa. Uusi metriikka voidaan kuitenkin syöttää suoraan painamalla lisää-nappia. Vanhojen tietojen muuttaminen edellyttää luonnollisesti ainakin yhden metriikan lisäämistä järjestelmään.

- Syötetään kevollinen metriikan nimi, mallin, arvoalue, oletussyöte ja kuvaus
- Syötetään tyhjä metriikan nimi, arvoalue, oletussyöte ja kuvaus

- Asetetaan maksimiarvo minimiarvoa pienemmäksi
- Asetetaan maksimiarvo ja minimiarvo samoiksi

### **VH3 Metriikkasarjojen tietojen muokkaaminen**

Vanhan metriikkasarjan muuttamiseksi on siirryttävä välilehdelle XXX ja valittava sarja. Uusi metriikka syötetään painamalla välilehden lisää-nappia.

- Syötetään kelvolliset epätyhjät tiedostot
- Syötetään tyhjät tiedot
- Annetaan käytössä oleva nimi metriikkasarjalle

### **VH4 Käyttäjätietojen muokkaminen**

Käyttäjätietojen muokkaaminen on mahdollista välilehdellä XXX. Uusia käyttäjiä voidaan lisätä lisää-napin painamisen jälkeen avautuvalla lomakkeella. Vanhojen tietojen muuttaminen tapahtuu valitsemalla käyttäjä, painamalla muokkaa-nappia ja muokkaamalla ruudulle piirrettävän esitetytyn lomakkeen kenttiä. Lopuksi painetaan päivitä-nappia.

- Annetaan kelvollinen nimi, sähköpostiosoite, puhelinnumero, käyttäjätunnus, salasana ja käyttöoikeudet