

## TIETOKANTA MERIKOTKIEN SEURANTAAN

Yhteenvetodokumentti

Versiohistoria:

Versio	Päivämäärä	Kuvaus	Tekijä
1.0	13.12.2007	Virallinen versio	Janne Piippo

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos  
Ohjelmistotuotantoprojekti Merikotka

Projektiryhmä:  
Janne Piippo  
Juha Hiekkämäki  
Pekka Maksimainen  
Petri Setälä  
Teemu Pulkkinen  
Tuire Huhtamäki

Ohjaaja:  
Sanna Keskiöja

## Sisällysluettelo

1. Johdanto .....	1
2. Lopputuotteen esittely .....	1
3. Dokumenttien tiivistelmät .....	1
4. Projektin päättöanalyysi .....	2
5. Yhteenveto .....	3

# 1. Johdanto

Yhteenvedodokumentti kokoaa kaikki toteutetun Haliaeetus-järjestelmän dokumentit yhteen, ja se on tarkoitettu ensimmäiseksi asiaksi jonka projektiin tutustuva henkilö lukee. Dokumentista löytyy lyhyt esittely tuotteesta, yhteenvedo muiden dokumenttien sisältämistä asioista sekä projektin päättöanalyysi. Päättöanalyysissä todetaan mikä onnistui ja mikä meni pieleen.

## 2. Lopputuotteen esittely

Haliaeetus- järjestelmä on merikotkan pesimätietojen seurantaan ja ylläpitoon kehitetty ohjelmisto, jonka käyttäjinä toimivat WWF:n merikotkatyöryhmän valtuuttamat henkilöt. Järjestelmän avulla kentältä tulevat pesätarkastustiedot tallennetaan tietokantaan WWW -käyttöliittymän kautta. Tietokantatietoja voidaan tarkastella kyseisen WWW -käyttöliittymän avulla tai vaihtoehtoisesti tuottaa niistä erilaisia raportteja. Käsiteltävä tieto sisältää pesimätietojen lisäksi myös kuvadataa. Järjestelmää on aiemmin toteuttanut kaksi ohjelmistotuotantoryhmää sekä itsenäisesti Henri Jääskeläinen että Jussi Ylikoski.

## 3. Dokumenttien tiivistelmät

### Vaatusmäärittelydokumentti

Vaatusmäärittelydokumentti sisältää asiakkaan ja ohjelmistotuotantoryhmä Merikotkan yhteisesti sopimat vaatimukset, jotka ohjelmistotuotantoryhmä Merikotkan tuli Haliaeetus-järjestelmään toteuttaa. Käyttäjävaatimuksista on priorisoidusti johdettu järjestelmävaatimukset suunnitteluvaihetta varten. Lisäksi dokumentti kuvaa järjestelmän tietosisällön sekä arkkitehtuurin yleisellä tasolla olemassa olevaan järjestelmään pohjautuen.

### Suunnitteludokumentti

Suunnitteludokumentti kuvaa toteutettavan Haliaeetus-järjestelmän teknisen toteutuksen näkökulmasta. Dokumentin perusteella kuka tahansa teknisesti osaava henkilö voisi toteuttaa kuvatun järjestelmän. Suunnitteludokumentti on siis ohje siitä, kuinka järjestelmä tulisi toteuttaa. Siinä kuvataan järjestelmän luokkarakenne, käyttöliittymä, tietokanta sekä järjestelmävaatimuksista johdetut tarvittavat muutokset.

### Testaussuunnitelma

Testaussuunnitelma sisältää tiedot siitä, miten toteutettavan järjestelmän testaus tehdään. Testaussuunnitelma on jaettu kolmeen vaiheeseen: yksikkötestaukseen, integrointitestaukseen sekä järjestelmätestaukseen. Testaussuunnitelmassa selvitetään kustakin testausvaiheesta lähestymistapa eli miten testataan, testattavat kohteet eli mitä testataan ja hyväksymiskriteerit eli milloin on testattu tarpeeksi kattavasti.

## **Ylläpitodokumentti**

Ylläpitodokumentti käsittelee sellaisia asioita, joita mahdolliset jatkoryhmät tai asiakkaan edustajat tarvitsevat muokatakseen Haliaeetus-järjestelmää. Dokumenttiin on listattu toteutusvaiheessa ilmenneitä asioita, jotka ovat joko muuttuneet suunnitteluvaiheesta tai muutoin vaikuttavat ohjelman ylläpitoon. Ylläpitodokumenttia ei ole tarkoitettu luettavaksi yksinään, vaan yhdessä suunnitteludokumentin ja ohjelmakoodin kanssa.

## **Käyttöohje**

Käyttöohje sisältää tietokantajärjestelmän uusien ja muuttuneiden ominaisuuksien kuvaukset ja käyttöohjeet. Järjestelmän entisellään säilyneitä toimintoja ei dokumentissa käsitellä, niiltä osin toimintaohjeet löytyvät Hali2-työryhmän 24.5.2004 päivätystä käyttöohjeesta. Dokumentti sisältää sekä ohjeistuksen järjestelmän käyttöön otosta että kuvitetut käyttöohjeet uusista ominaisuuksista.

## **4. Projektin päättöanalyysi**

Toteutusvaihetta auttoi kunnollinen versionhallinta ja sen sisältämien tiedostojen automaattinen kopiointi ajettavaan järjestelmään. Näin oleellinen toiminnallisuus saatiin erotettua ajoympäristön epäoleellisista yksityiskohdista ja asetuksista. Dokumentoinnissa Wiki-järjestelmä helpotti työskentelyä ja säästi lukuisia tunteja aikaa dokumentaation synkronoinnissa ja editoinnin helppoudessa.

Projektin aikataulut onnistui riittävän hyvin. Kaikki välttämättömiksi luokitellut ominaisuudet sekä joitain alemman prioriteetin asioita saatiin toteutettua ilman loppupaniikkia. Yksittäisten tehtävien aika-arviot eivät aina pitäneet, mutta ongelmista selvittiin kokonaisaika-aulun puitteissa. Prioriteetteja ei määrittelyvaiheen jälkeen täytynyt muokata ja suunnitteludokumentti oli tarpeeksi kattava toteutusvaihetta varten. Suunnitteluvaiheessa tosin osa ryhmästä tutki edellisen ohjelmistotuotantoprojektin järjestelmää, joka ei sisältänyt viimeisimpiä muutoksia. Asia huomattiin vasta toteutusvaiheen alussa. Koska rakenne korkealla tasolla ei kuitenkaan ollut muuttunut, ei tämä aiheuttanut merkittäviä ongelmia.

Aiempien ryhmien materiaali oli välillä harhaanjohtavaa. Suunnitteluvaiheessa edettiin aluksi olettaen, että edellisen ryhmän suunnitteludokumenttaatio kuvaisi olemassa olevaa järjestelmää. Tämä ei kuitenkaan pitänyt paikkaansa, sillä osa kuvatuista osajärjestelmistä jopa puuttui kokonaan. Puutteellisen loppuraportin ja edellisen ryhmän jälkeen tehtyjen dokumenttoimattomien muutosten vuoksi monia yksityiskohtia piti kaivaa suoraan lähdekooditasolta. Lisäksi järjestelmän koodikonventioissa oli suuria heittoa, jotka vaikeuttivat luettavuutta ja osaltaan hidastivat työtä.

Olemassa olevan järjestelmän rakenne ei ollut paras mahdollinen. Todellisen modulaarisuuden puuttuessa sitä voidaan käsitellä vain kokonaisuutena, joka toimii ainoastaan db-palvelimen ajoympäristössä. Tämä hankaloitti toteutusvaihetta, sillä emme keksineet miten Javan virheenetsintätyökaluja olisi voinut ajettavaan servlettiin käyttää. Näin ollen ainoaksi tavaksi seurata koodin suoritusta jäi erilaisten viestien tulostaminen logitiedostoon. Useimmissa virhetilanteissa järjestelmä ei antanut mitään palautetta ja selaimen ruudulle ilmestyi pelkkä tyhjä sivu ilman syytä selvittäviä logiviestejä. Paransimme mallipohjätiedostojen virheiden raportointia, mutta esimerkiksi monissa poikkeustapauksissa ei järjestelmä edelleenkään tulosta auttavaa informaatiota. Koska järjestelmää voi kehittää vain db-palvelimella, jouduimme käyttämään lukkoskriptiä rajoittaaksemme muutosoikeuden vain

yhdelle käyttäjälle kerrallaan. Tämän vuoksi järjestelmää pystyi muokkaamaan vain yksi henkilö kerrallaan. Lukkokäytäntö hidasti hieman kehitystä, mutta toisaalta vältti samanlaisesta editoinnista aiheutuvia mahdollisia virheitä.

Järjestelmä ei sisältänyt yksikkötestejä, mikä hankaloitti muutosten toimivuuden varmistamista. Rakenteen puutteiden vuoksi emme yksikkötestejä itsekään voineet lisätä.

## 5. Yhteenveto

Testauksen järjestäminen ohjelmistotuotantoprojektin yhteydessä on nykyisellään liian haastavaa, uusia opeteltavia asioita ja työkaluja on muutenkin tarpeeksi. Projektin yhteydessä pidetty parin tunnin perehdyttämisluento testauksesta ei antanut riittävää pohjaa testauksen kunnolliseen toteuttamiseen. Testaus on kurssin isoista vastualueista ainoa, josta ei ole perus- ja aineopintojen yhteydessä minkäänlaista opetustarjontaa. Aihealueeseen perehtyminen projektin ohella on tietysti palkitsevaa, mutta projektin aikataulun puitteissa hankalaa. Lopussa tiesi, mitä alussa olisi kannattanut tehdä. Terveisiä kurssisuunnitteluun - kirjallisuuden mukaan testaukseen käytettävän ajan suositellaan olevan 25-50 % koko ohjelmistokehityksen aikaresursseista, ylläpito pois lukien. Tämä tekee siitä tärkeän perus- ja aineopintoihin kuuluvan aiheen.

Versionhallinnan tärkeyttä ei kurssin yhteydessä tarpeeksi korosteta. Projekteissa useamman henkilön muokkaukset samaan järjestelmään täytyisi pitää hallinnassa, mutta mitään kunnollista ohjetta tähän ei ole. Merikotkaprojektissa ongelmia ei ryhmän alan tuntemuksen vuoksi ilmennyt, mutta kurssin pääsivua pitäisi aiheesta päivittää. "Yhteenveto CVS -komennoista" on sekava eikä dokumentin nimestä tai sisällöstä käy edes ilmi, että kyseessä on ohje versionhallintaan. Nykyisellään oleellinen asia jää ryhmän jäsenten harastuneisuuden varaan. Kurssin resursseihin kannattaisi myös lisätä ohje Wiki –järjestelmän käyttöönotosta. Wikin voivat kaikki opiskelijat pyytää palvelimelta "wiki.helsinki.fi". Wikiä ei siis tarvitse asentaa itse ja onkin vaikea keksiä syitä sille, miksi sitä ei projekteissa käytettäisi.

Projektin aikana aikaa kului paljon olemassa olevan järjestelmän hahmottamiseen sekä työkaluihin kuten kehitysympäristö, versionhallinta, debuggaus-menetelmät ja Freemarker perehtymiseen. Ryhmässä jokainen teki osansa ja apua saatiin ja annettiin tarpeen vaatiessa. Muutaman ryhmäläisen erityisosaamisen ansiosta mm. versionhallinta ja pari hankalaa käyttöliittymätoteutusta sujuivat kiitettävän tehokkaasti. Kurssin suurin ansio oli siinä, että sen aikana joutui hyödyntämään soveltuvin osin kaikkea tähän asti opittua ja sen pohjalta etsimään uutta tietoa. Jo olemassa olevan ja hyödyllisen järjestelmän kehitystyö oli kiinnostavaa ja opettavaa, joskin myös haastavaa ja ajoin turhauttavaa. Motivaatiota paransi järjestelmän merkitys tieteellisessä tutkimuksessa. Enemmän tästä oppi kuin alusta asti itse tehdystä projektista. Kokonaisuudessaan projekti oli onnistunut, sillä kaikki korkeimman ja osa alemman prioriteetin ominaisuuksista saatiin toteutettua ilman loppupaniikkia.