



# **WEB-POHJAISET PROJEKTIINHALLINTASOVELLUKSET**

**Esa-Pekka Lahnalampi  
Jani Malkamäki**

**Opinnäytetyö  
Huhtikuu 2008**

**Liiketalous**



**JYVÄSKYLÄN  
AMMATTIKORKEAKOULU**

Tekijä(t) LAHNALAMPI, Esa-Pekka  MALKAMÄKI, Jani	Julkaisun laji Opinnäytetyö	
	Sivumäärä 75	Julkaisun kieli Suomi
	Luottamuksellisuus <input type="checkbox"/> Salainen _____ saakka	
Työn nimi WEB-POHJAISET PROJEKTINHALLINTASOVELLUKSET		
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma		
Työn ohjaaja(t) KIVIAHO, Niko		
Toimeksiantaja(t) Jyväskylän ammattikorkeakoulu		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli tutkia web-pohjaisia projektinhallintasovelluksia, ja niiden soveltuvuutta yritysten käyttöön. Projektinhallintasovelluksista pyrittiin etsimään projektien vaatimat ominaisuudet ja toiminnallisuudet. Sovelluksia testattiin niiden ominaisuuksien ja käytettävyyden näkökulmasta. Työn tuloksena on testausraportti kustakin testatusta sovelluksesta. Sovellusten ominaisuudet koottiin helppolukaiseen ominaisuusluetteloon.</p> <p>Teoria pohjautui projektinhallintaan sen vaiheiden perusteella. Lisäksi teoriaosiossa käsiteltiin millaisia vaatimuksia web-pohjainen projektinhallintasovellus vaatii yrityksen intranetin infrastruktuurilta. Teoria on koottu projektinhallintakirjallisuudesta ja IT-kirjallisuudesta. Työssä on käytetty myös paljon internetlähteitä työn luonteen takia. Teoria muodosti pohjan testattavien sovellusten testauskriteereille ja projektin asettamille vaatimuksille. Osa tutkimuspohjasta muotoutui omien kokemusten perusteella.</p> <p>Opinnäytetyöhön on listattu eri sovellusten ominaisuudet sekä kirjallinen testiraportti kustakin testauksessa mukana olleesta sovelluksesta. Sovellustestauksessa pyrittiin miettimään, miten sovellus toimisi projektinhallinnan työkaluna ja olisiko se käyttökelpoinen. Sovelluksen helppokäyttöisyys oli testauksessa suuressa roolissa. Myös sovelluksen toimivuuteen erilaisissa ympäristöissä kiinnitettiin huomiota.</p> <p>Sovellustestaus suoritettiin tekemällä kevyt testaussuunnitelma, jota seurattiin jokaisessa testitapahtumassa testauksen aikana. Kunkin sovelluksen testiraporttiin kirjattiin myös sovelluksen tarjoamat erityisominaisuudet ja puutteet. Testauksessa jokaiselle sovellukselle pystytettiin testiympäristö sovelluksen vaatimusten perusteella.</p>		
Avainsanat (asiasanat) projektinhallinta, web-sovellus, käytettävyys, projektinhallintasovellus, intranet, tietoturva		
Muut tiedot		

Author(s) LAHNALAMPI, Esa-Pekka  MALKAMÄKI, Jani	Type of Publication Bachelor's Thesis	
	Pages 75	Language Finnish
	Confidential <input type="checkbox"/> Until _____	
Title WEB-BASED PROJECT MANAGEMENT APPLICATIONS		
Degree Programme Business Information Systems		
Tutor(s) KIVIAHO, Niko		
Assigned by The school of business and administration		
Abstract  The objective of the thesis was to research web-based project management applications and their suitability for projects. The aim was to find features and functionalities that project management usually demands. The applications were tested in standpoint of features and usability. The results of the thesis are short written test reports of every application. Application features were assembled in a feature list.  The theory part of the thesis is mainly founded on the theory of project management. The theory section also examines what kind of specific needs a web-based application sets for company intranet infrastructure. The theoretical part of the study was compiled from project management and IT literature. Because of the nature of the thesis, it has several web-based information sources. The theory used in the thesis acts as the base for the software test criteria. The other observation method for application testing is partly reconstructed from our own experiences from software testing and building applications.  In application testing the main goal was to figure out, how the application would work on project management tool and what kind of features and functionality the project demands from the application. Usability acted big role at the test process. Also application functionality in different project types was observed.  The application testing was carried out by constructing a test plan and executed by following test plan contents. A list of the strengths and weaknesses of the program was built from the application testing results. The test environment was built-up individually for every tested application.		
Keywords project management, web application, usability, project management application, intranet, security		

# SISÄLTÖ

<b>1 JOHDANTO</b> .....	<b>5</b>
<b>2 TUTKIMUSASETELMA</b> .....	<b>6</b>
2.1 Opinnäytetyön taustateoria, tavoitteet ja rajaukset.....	6
2.2 Tutkimusmenetelmät.....	7
2.3 Tutkimuskysymykset .....	7
<b>3 PROJEKTINHALLINTA</b> .....	<b>9</b>
3.1 Projekti käsitteenä .....	9
3.2 Projektin elinkaari .....	10
3.3 Käynnistysvaihe (initiation).....	11
3.4 Suunnitteluvaihe (planning).....	16
3.5 Suoritus- ja kontrollointivaihe (execution).....	20
3.6 Lopetusvaihe (closure) .....	21
<b>4 WEB-SOVELLUSTEN TOIMINTA JA RISKIT</b> .....	<b>23</b>
4.1 Yleistä .....	23
4.2 Web-palvelin .....	23
4.3 Web-sovellusten alustat .....	25
4.4 Web-tekniikat.....	26
4.5 Web-ympäristön riskit.....	28
<b>5 KÄYTETTÄVYYS</b> .....	<b>33</b>
5.1 Käytettävyys web-sovelluksessa.....	33
5.2 Käytettävyyden mittaaminen .....	35
5.3 Sovellettava menetelmä.....	35
<b>6 TESTATTAVAT PROJEKTINHALLINTASOVELLUKSET</b> .....	<b>37</b>
6.1 Alustavat toimenpiteet .....	37
6.2 Easy Projects .NET .....	40
6.3 TinyErp .....	45
6.4 ActiveCollab.....	50
6.5 TRAC.....	54
6.6 SugarCRM .....	59
6.7 dotProject.....	63
<b>7 TULOSTEN YLEINEN TARKASTELU JA ARVIOINTI</b> .....	<b>68</b>
7.1 Tutkimuksen tulokset .....	68
7.2 Jatkotutkimusmahdollisuuksia .....	69
7.3 Pohdintaa.....	69
<b>LÄHTEET</b> .....	<b>71</b>
<b>LIITTEET</b> .....	<b>1</b>

## KUVIOT

KUVIO 1. Projektin elinkaari.....	10
KUVIO 2. Riskianalyysikaavio (Project perfect n.d.) .....	14
KUVIO 3. Easy Projects .NET, käyttäjän etusivu.....	42
KUVIO 4. Easy Project .NET, kalenterinäkymä.....	43
KUVIO 5. Easy Project .NET, kustomoitujen raporttien luontilomake.....	44
KUVIO 6. TinyErp-projektintehtävälistaus .....	47
KUVIO 7. TinyErp, projektinäkymä.....	48
KUVIO 8. ActiveCollab-sovelluksen kojelauta .....	52
KUVIO 9. ActiveCollab-sovelluksen kalenteri .....	53
KUVIO 10. Trac-sovelluksen etusivu .....	56
KUVIO 11. Trac-sovelluksen tehtävän määrittäminen.....	57
KUVIO 12. Trac-sovelluksen aikajana.....	57
KUVIO 13. Trac-sovelluksen roadmap .....	58
KUVIO 14. Trac-sovelluksen raportit.....	58
KUVIO 15. SugarCRM projektivälilehti alkunäkymä.....	61
KUVIO 16. SugarCRM uuden projektin luominen .....	62
KUVIO 17. SugarCRM, Gantt -kaavio.....	62
KUVIO 18. DotProject, projektinäkymä.....	66
KUVIO 19. DotProject, Gantt -kaavio.....	66

## LIITTEET

LIITE 1. Sovellusten ominaisuusvertailutaulukko. ....	1
LIITE 2. Sovellusten käytettävyyssarviotaulukko.....	3

## KÄYTETYT LYHENTEET JA KÄSITTEET

Käsite	Selostus
AJAX (Asynchronous JavaScript And XML)	Tekniikka vuorovaikutteisten verkkosovellusten luomiseen
Apache	Avoimeen lähdekoodin perustuva HTTP-palvelinsovellus
Arkkitehtuuri	Järjestelmän yleinen rakenne-peruste
ASP (Application service provider)	Sovelluksen vuokraaja
Autentikointi (authentication)	Autentikoinnilla tarkoitetaan yleisesti käyttäjän henkilöllisyyden varmentamista.
Avoin lähdekoodi (Open source)	Sovellukset joita kuka tahansa voi korjata, kehittää ja kopioida.
C/C++, C#	Olio-ohjelmointikieliä
CRM (Customer relationship management)	Asiakkuudenhallinta-tietojärjestelmä.
CSS (Cascading Style Sheets)	Kaskadisit tyyliohjeet. Erityisesti WWW-dokumenteille kehitetty kaskadinen tyyliohjejärjestelmä.
ERP (Enterprise resource planning)	Yrityksen tietojärjestelmä, joka integroi eri toimintoja kokonaisuuksiksi.
Gantt-kaavio	Gantt-kaavio kuvaa tehtäviä aikajanalla niiden ajoitusta osoittavien pylväiden avulla.
HTML (Hypertext Markup Language)	Avoimesti standartoitu kuvauskieli. HTML tunnetaan erityisesti web-sivustojen kuvauskielenä.
HTTP (Hypertext Transfer Protocol)	Hypertekstin siirtoprotokolla
HTTPS (Hypertext Transfer Protocol Secure)	HTTP-protokollan salattu versio
IIS (Internet Information Services)	Microsoftin kehittämä palvelohjelmistokokonaisuus, joka on tarkoitettu käytettäväksi Windows-palvelimissa.
Intranet	Lähiverkko, joka eristetty tietyn ryhmän käyttöön. Yleensä yrityksissä sisäinen verkko.
Java	Teknologiaperhe, ohjelmistoalusta, johon kuuluu oliopohjainen ohjelmointikieli.
Linux	Käyttöjärjestelmä, joka koostuu Linux-ytimeistä.

Mac OS X	Applen kehittämä ja markkinoima käyttöjärjestelmä Macintosh-tietokoneisiin
MIME (Multipurpose Internet Mail Extension)	MIME määrittelee välitetyn datan muodon.
MySQL, PostgreSQL, MS SQL	Tietokannan hallintajärjestelmiä.
NTFS, FAT32, FAT	Tiedostojärjestelmiä
Ohjelmistokehys (framework)	Muodostaa rungon sen päälle rakennettavalle sovellukselle.
Organisaatio	Hallinnollinen tai toiminnallinen kokonaisuus
PDF (Portable Document Format)	Adoben kehittämä PostScript-kieleen pohjautuva siirrettävä tiedostomuoto
PHP (Hypertext preprocessor)	Ohjelmointikieli, jota käytetään erityisesti web-palvelinympäristön dynaamisten web-sivujen luonnissa.
Projektipäällikkö	Henkilö, joka on yrityksessä vastuussa yhdestä tai useammasta projektista. Myös käyttäjärooli, jolla on projektipäällikön oikeudet projektiin.
Proseduuri	Määritelmä sarjasta toimintoja tai tehtäviä
Raportti	Tiedonanto, selonteko, tav. kirjallinen melko yksityiskohtainen tilannekatsaus
Resurssi	Voimavara, toiminnan edellytys (raha, tila, työvoima jne.).
Rooli	Järjestelmässä tai ryhmässä käyttäjän rooli
Solaris (Solaris Operating Environment)	Sun Microsystemsin kaupallinen käyttöjärjestelmä.
sovellus	Ajettavissa oleva sovellus. Toteutettu jollakin ohjelmointikielellä.
SQL (Structured Query Language)	IBM:n kehittämä standartoitu kyselykieli, jolla relaatiotietokantaan voidaan tehdä erilaisia hakuja.
Unix	Laitteistoriippumaton käyttöjärjestelmä
Web (WWW)	Internetissä toimiva hajautettu hypertekstijärjestelmä
Windows	Käyttöjärjestelmä, maailman käytetyin käyttöjärjestelmä
WWW (Word wide web)	Internetissä toimiva hajautettu hypertekstijärjestelmä
XML (eXtensible Markup Language)	Merkintäkieli, jolla tiedon merkitys on kuvattavissa tiedon sekaan.

## 1 JOHDANTO

Projektien ajankäytön ja edistymisen seuraaminen on tärkeää projektien parissa työskenteleville yrityksille. Projektien kannalta on ensiarvoisen tärkeää tiedostaa projektin nykyinen tila, projektiin käytetyt tunnit, projektin taloudellinen tilanne sekä projektiin liittyvät tehtävät. Työtehtävien suunnittelu ja jakaminen projektiryhmälle on osa yrityksen projektinhallintaa.

Projektien seurantaan helpottavia työkaluja on saatavilla useita. Työkalut kulkevat yleensä nimellä projektinhallinta (Project management), ERP (Enterprise resource planning), EPM (Enterprise project management) ja BI (Business intelligence). Projektinhallintaohjelmistojen funktio on auttaa projektiryhmää projektinseurannassa ja aikataulutuksessa.

Pienten kasvavien yritysten ongelmaksi muodostuu usein projektinhallintajärjestelmän puuttuminen, tai sen riittämättömyys vastata yrityksen kasvaviin tarpeisiin. Etätyön määrän lisääntyminen aiheuttaa yrityksillä uudenlaisia tarpeita projektihenkilöstön ja sen sidosryhmien väliseen kommunikointiin.

Yrityksen projektinhallintaohjelmisto on usein vain Excel-taulukko, joka on jaettu verkkoaseman välityksellä useille työasemille. Tällainen järjestely luo suuren riskin tiedon katoamiseen, tai korruptoitumiseen. Tietokanta on turvallisempi tapa säilyttää tietoa verrattuna Excel-tiedostoon.

Tutkimuksen avulla selvitetään miten erilaiset projektinhallintasovellukset vastaavat projektien vaatimuksiin. Työssä käydään läpi erityyppisiä projektinhallintasovelluksia. Sovelluksista kartoitetaan soveltuvuus projektinhallintatyökaluksi, ja se, kuinka käytettävä kyseinen sovellus todellisuudessa on.

Tutkimuksessa käydään läpi myös yleisimpiä web-tekniikan aiheuttamia riskejä ja vaatimuksia. Web-ympäristössä toimiva sovellus on helposti avoimena erilaisille hyökkäyksille, mikä lisää tietoturvariskejä.



## 2 TUTKIMUSASETELMA

Luvussa käydään läpi tutkimuksen taustateoria, rakenne, rajaukset ja tutkimuskysymykset.

### 2.1 Opinnäytetyön taustateoria, tavoitteet ja rajaukset

Taustateorianaan toimivat projektinhallintaan kytkeytyvä kirjallisuus ja vertailtavien projektinhallintaohjelmistojen omat dokumentaatiot. Projektinhallintakirjallisuutta hyödynnetään osana arvosteluperusteiden laadintaa. Sovelluksien testaamisesta saatujen kokemusten kautta havainnoidaan sovellusten käytettävyyttä ja toiminnallisuutta.

Päätavoitteena opinnäytetyöllä on tuottaa yleissivistävä katsaus web-pohjaisten projektinhallintaohjelmistojen ominaisuuksista ja käytettävyydestä projektinhallintatyökaluna.

Opinnäytetyössä tapahtuvaan testaukseen valitaan projektinhallintasovellukset. Sovelluksien minimi vaatimukset ovat seuraavat:

- Sovelluksen on kyettävä käsittelemään useampaa kuin yhtä projektia.
- Sovelluksen on toimittava web-ympäristössä.
- Sovelluskehityksen taustalla toimii aktiivinen yhteisö.
- Käyttöönoton kustannusarvio, käyttöönotosta ei pidä aiheutua kohuttomia kustannuksia asiakkaalle.
- Sovellus on mahdollista asentaa paikallisesti, eli ei ohjelmistovuokrausta (Application Service Provider, ASP).

Rajauksen tarkoitus on luoda tilanne, jossa vertailtavat ohjelmistot ovat keskenään mahdollisimman samankaltaisia.

Ensimmäisen rajauksen tarkoituksena on rajata vertailusta pois sovellukset, jotka soveltuvat vain yksittäisen projektin hallintaan.

Toisessa rajauksessa rajataan pois työpöytäkäyttöön tarkoitettut sovellukset, sillä vertailussa on mukana ainoastaan web-ympäristössä toimivia sovelluksia.

Kolmannessa kohdassa rajataan pois sovellukset jotka ovat ns. "kuolleita" sovelluksia. Kuolleiksi sovelluksiksi lasketaan sovellukset, joiden kehitys on lopetettu. Kuolleet sovellukset ovat hyvin tyypillisiä OSS-yhteisöissä (Open Source Software Community).

Neljännän rajauksen tarkoituksena on rajata pois sovellukset, jotka aiheuttavat suuria investointeja kohdeyritykselle. Kustannukset sisältävät mahdolliset sovelluslisenssit ja palvelin-kustannukset.

Viides kohta rajaa pois sovellukset, joita tarjotaan vain ASP-lisenssin alaisena. Tavoitteena on löytää sovelluksia, jotka ovat asennettavissa paikallisesti yrityksen sisäiseen verkkoon (intranet).

## 2.2 Tutkimusmenetelmät

Tutkimusmenetelmänä tutkimuksessa on konstruktiiivinen tutkimusote. Konstruktiiivisen tutkimuksen lähtökohta on kohdeyritysten ongelma projektihallinnan uudelleen organisoimisesta. Ongelmaan pyritään kehittämään ratkaisukonstruktio, joka perustuu osittain olemassa olevaan tietämykseen ja osittain tutkimusprosessiin. Opinnäytetyön tavoitteena on helpottaa yritysten projektinhallintasovellusten valintaa.

## 2.3 Tutkimuskysymykset

Alla luettelo tutkimuksessa käytetyistä tutkimuskysymyksistä.

1. Millaisia ominaisuusvaatimuksia projektinhallinta asettaa projektinhallintasovellukselle?
2. Mitä erityisvaatimuksia web-pohjaisen projektinhallintasovelluksen käyttöönotto asettaa?
3. Mitä riskejä web-pohjainen projektinhallintasovellus, sekä sen käyttämät web-tekniikat voivat aiheuttaa?

Ensimmäinen tutkimuskysymys pohtii projektinhallinnan asettamia vaatimuksia projektinhallintasovellukselle. Kysymykseen vastausta etsitään projektinhallintaan liittyvästä teoriasta. Projektille tärkeät seurannan kohteet karotetaan ja mietitään miten tutkittavat sovellukset vastaavat todettuihin vaatimuksiin.

Toisessa tutkimuskysymyksessä mietitään millaisia erityisvaatimuksia web-pohjaisen projektinhallintasovelluksen käyttöönotto ja käyttö asettaa. Vastaus etsitään lähdemateriaaleista sekä web-sovellusten käyttöönoton aikana suorittavasta havainnoinnista.

Kolmannessa kysymyksessä pohditaan web-pohjaisten sovellusten yleisimpiä riskejä. Kysymykseen etsitään vastausta käytännön testeistä ja lähdemateriaalista.

### 3 PROJEKTIHALLINTA

Tässä luvussa tutustutaan projektinhallintaan terminä sekä projekteihin ja projektinhallintaan liittyviin piirteisiin. Luvussa käydään läpi projektin elinkaari, projektin vaiheet ja riskit, sekä riskien hallinta.

#### 3.1 Projekti käsitteenä

Projektinhallintasovellukset ovat projektien aikatauluttamisen, resursoinnin ja osittamisen työkaluja. Projektinhallintasovellukset eivät toimi yrityksen kokonaisvaltaisena tietojärjestelmänä.

Projektinhallinnan teorian pohjalta on pääteltävissä, että projektinhallintasovelluksen tulisi sisältää ainakin seuraavat ominaisuudet:

- projektin aikataulutus
- projektiryhmän roolitukset projektissa ja sovelluksessa,
- raportointiominaisuudet
- projektin tehtävien jaottelu ja jaksottaminen
- projektin resursointi-ominaisuudet
- riskien hallinta
- ongelmien hallinta
- sovelluksen kollaboraation ominaisuudet.

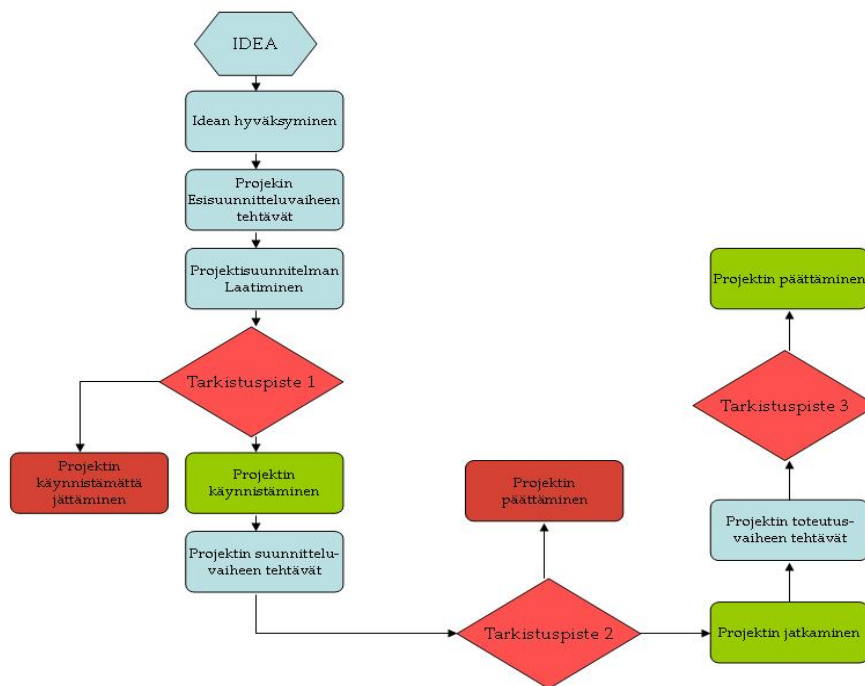
Tarkoituksen on valmistella tutkimuksessa toteutettavaa projektinhallintasovellusten vertailua ja selvittää miten erityyppiset projektinhallinnan teoriat vaikuttavat sovelluksille oleellisten ominaisuuksien hahmottamiseen.

Virtanen (2000, 35) määrittelee projektin terminä seuraavasti. Projekti on kertaluonteinen toimenpide tai hanke, jolla on tietty organisaatio ja resurssit, joka

pyrkii toteuttamaan tiettyä ennalta sovittua tehtävää ja joka etenee tiettyjen kehysvaiheiden (suunnittelu, käynnistäminen, toimeenpano, ohjaus ja arviointi) mukaan sen perusteella, mikä hankkeen tehtäväksi on määritelty.

### 3.2 Projektin elinkaari

Projekti on ajallisesti rajattu, eli projektilla on selkeä alkamis- ja päättymisajankohta. Projektin elinkaari jakautuu eri vaiheisiin, projektin käynnistämistä sen päätösvaiheeseen. Elinkaari sisältää koko projektin elinajan, ensimmäisestä tehtävästä viimeiseen. Kuvio 1 havainnoi projektin elinkaaren rakenteen.



KUVIO 1. Projektin elinkaari

Riippuen projektin monimutkaisuudesta, projektissa käydään läpi useita tehtäväsarjoja sen elinkaaren aikana. Vaiheet ovat muodollisia vaiheita. Vaiheet voivat vaihdella projektikohtaisesti. (Project perfect n.d.)

### 3.3 Käynnistysvaihe (initiation)

Ensimmäisenä vaiheena voidaan pitää projektin käynnistysvaihetta. Käynnistysvaiheessa tutkitaan mitä teknologiaa ja millaista työryhmää projektin suorittamiseen tarvitaan. Projektipäällikkö päätetään yleensä tässä vaiheessa. Projektipäällikön tehtäviin kuuluu projektiryhmän kokoaminen ja organisointi. (Project perfect. n.d.)

Käynnistysvaiheessa mietitään myös projektin tarpeellisuus ja ajankohtaisuus. Käynnistysvaihe saa usein alkunsa ideasta, joka voidaan toteuttaa projektiluontoisena työskentelynä. Projektilla on aina päämäärä, joka pyritään saavuttamaan. Ennen kuin projekti aloitetaan, on hyvä löytää vastaus ainakin seuraaviin kysymyksiin (Project perfect n.d.):

- Ketkä ovat avainryhmässä ja mikä avainryhmän rooli on?
- Mikä heidän tehtävänsä on ja mistä koko projektissa on kyse?
- Ovat he valmiita antamaan panoksensa projektiin?
- Mikä heidän odotuksensa on lopputuloksesta?

Projektin käynnistysvaiheen jälkeen on selvillä, ketkä projektissa työskentelevät, kuka projektin maksaa, mitä projektin lopputulokselta odotetaan ja kuinka kauan lopputuloksen saavuttaminen vie aikaa. Useiden projektien epäonnistumisen syy on käynnistysvaiheen heikko toteutus, joka usein aiheuttaa projektisuunnitelman epäonnistumisen. (Project perfect n.d.)

Ennen projektisuunnitelmaa tehdään projektiehdotus. Projektiehdotus toimii esiselvityksenä suunnittelun aloittamiselle, mikä kuuluu myös käynnistysvaiheen tehtäviin. Projektiehdotuksessa analysoidaan ongelmat ja tarpeet, joiden

pohjalta projektisuunnitelma ja projekti saavat alkunsa. Projektiehdotuksen jälkeen suunnitelman toteuttamiseen otetaan mukaan projektityöryhmä ja kaikki sidosryhmät joita projekti koskee. (Projektisuunnitelma n.d.)

Seuraavia asioita voidaan pitää hyvän projektisuunnitelman pohjana: (Projektisuunnitelma n.d.)

- Tavoiteasettelun pitää olla selkeä ja realistinen. Hankkeen tärkeimmät tulokset ja toteutusmalli voidaan johtaa tavoitteenasettelusta.
- Projekti aikataulun on oltava selkeästi ja realistisesti määritelty.
- Projektin johtamismalli pitää olla johdonmukaisesti määritelty. Se edellyttää projektin eri osapuolien roolien, vastuiden, resurssien sekä projektin toteutusorganisaation sekä päätöksenteko- ja raportointimallien selkeää määrittelyä.
- Projektin resurssit pitää määrittellä selkeästi ja niiden on oltava oikeassa suhteessa asetettuihin tavoitteisiin ja avaintuloksiin.

### 3.3.1 Projektin riskit

Jokaisessa projektissa on mahdollisia riskejä. Projektin riskejä selvitetään erilaisilla kartoitusmenetelmillä. Kartoituksen perusteella mietitään ratkaisumalleja mahdollisille toteutuville riskeille. Riskien toteutuessa valmis ratkaisumalli vähentää toteutuneen riskin aiheuttamaa haittaa projektille. Yleisiä syitä projektin epäonnistumiseen ovat seuraavat (Rissanen 2002, 12):

- Tavoitteita ei pohdita riittävästi
- Innovatiivisuus puuttuu
- Huono johtaminen ja vähäinen projektiryhmän motivointi.
- Projektityömenetelmä ei ole hallussa.
- Ulkopuolista asiantuntemusta ei hyödynnetä riittävästi projektissa.

- Projektin tuloksista ei tiedoteta edes asiakkaalle.

Riski on aina mahdollisuus, joten sen todennäköisyys tapahtua on alle 100 %. Jos jokin asia toteutuu varmasti, se ei ole riski, vaan ongelma (issue). Ongelmahallinta erotetaan riskihallinnasta ja sitä käsitellään erillisillä menetelmillä. (Project perfect n.d.)

Riskin ominaispiirre on kuitenkin aina mahdollisuus toteutua. Jos tapahtumisen mahdollisuus on nolla, se ei ole riski. Riski on myös aina vahingollinen, joten riskiä tunnistettaessa on aina punnittava sen vahingollisuus projektille ja mahdollisuus toteutua. (Project perfect.)

Riskien hallintasuunnitelmassa on neljä vaihetta. Vaiheet ovat (Project perfect n.d.):

- riskin tunnistaminen,
- riskien vakavuuden mittaaminen,
- riskiin varautuminen,
- riskien seuranta ja kontrollointi.

### 3.3.2 Riskien tunnistaminen

Hyvä keino riskien tunnistamiseen on koota työryhmä, jossa on edustajia jokaisesta osastosta, joka projektiryhmään kuuluu. Riskien erilaisuuden kannalta on hyödyllistä saada useita näkökulmia tunnistamaan mahdollisia riskejä. Jokainen osasto kehittää omat metodinsa riskien hallintaan ja niiden tunnistamiseen. Vaikutus voi olla taloudellinen, tai vaihtoehtoisesti esimerkiksi virsantäpölväs jää saavuttamatta. (Project perfect n.d.)

### 3.3.3 Riskien vakavuuden kartoittaminen

Riskit kartoitetaan kahden muuttujan perusteella. Ensimmäinen muuttuja on



riskin vakavuus. Riskin vakavuudella tarkoitetaan sen aiheuttamaa vaikutusta projektille. Toinen muuttuja on riskin todennäköisyys, jolla tarkoitetaan riskin todennäköisyyttä tapahtua. (Project perfect n.d.)

Yleensä kriittisesti vaikuttavat riskit ovat todennäköisyydeltään harvinaisempia verrattuna vaikutukseltaan alhaisiin riskeihin. Riskikartoitusta tehtäessä pahin riski on vakavuudeltaan kriittinen ja todennäköisesti tapahtuva. Riskien kriittisyyden mittausta voidaan kuvata seuraavalla kuviolla (Project perfect n.d.):

Todennäköisyys	4	Keskitaso	Kriittinen
	3		
	2	Alhainen	Korkea
	1		
		1	2
			3
			4
		Vaikutus	

KUVIO 2. Riskianalyysikaavio (Project perfect n.d.)

### 3.3.4 Riskiin varautuminen

Mahdolliseen riskiin voi varautua ennakkoon, kunhan se on ensin määritelty.

Yleisiä strategioita riskien vaikutuksen lieventämiseen ovat (Project perfect n.d.):

- riskin välttäminen ennen kuin se on tapahtunut

- riskin siirtäminen kolmannelle osapuolelle, esimerkiksi toiselle yritykselle
- riskin lieventäminen ennalta tehdyillä valmisteluilla
- riskin hyväksyminen.

### 3.3.5 Riskien kontrollointi

Viimeinen taso on riskien jatkuva seuranta. Riskien kartoituksen jälkeen riskejä seurataan koko projektin ajan. Valmiilla riskianalyysillä tunnistetaan mahdolliset toteutuvat riskit. (Project perfect n.d.)

### 3.3.6 Yleisimmät riskitekijät

Yksi yleisimmistä riskitekijöistä on huono projektisuunnitelma. Hyvin suunnitellussa projektissa osa riskeistä katoaa automaattisesti. Huonon projektisuunnitelman tunnusmerkkejä ovat seuraavat (Rissanen 2002.):

- projektit myöhässä sovitusta aikatauluista
- projektiryhmällä jatkuvia ylitöitä ja kiireitä
- jatkuva projektien ja projektityövoiman priorisoinnin tarve
- muutokset tulevat myöhässä ja niitä ei voida hallita
- projektit toimitetaan tilaajalle keskeneräisinä.

Projektisuunnitelma toimii viitekehyksenä projektille. Projektiryhmälle etäännyminen projektin tehtävästä on aina riski. Projektisuunnitelma ja hyvä esituskimus edesauttaa projektin pitämistä viitekehyksen sisällä. (Rissanen 2002.)

Lait ja säädökset aiheuttavat usealle projektille epäonnistumisen mahdollisuuden. Kyseiset riskit tulevat esiin projektin elinkaaren loppuvaiheessa. Tästä johtuen projektiryhmän onkin huomioitava mahdolliset muutokset laissa kansainvälisiä projekteja tehtäessä. (Cornford-Smithson 2006, 78 - 97.)

Riskitekijä, joka tulisi ottaa huomioon projektin alkuvaiheissa on projektin ainutlaatuisuus. Projektiryhmälle selviää usein projektin aikana, että kyseinen asia ei luo mitään uutta markkinoille. Projektin lähtökohdan on tarkoitus olla innovatiivinen ja kehittävä. Kyseinen riski voidaan ylittää tekemällä projektista entistä syvempi ja laajempi. (Cornford-Smithson 2006, 78 - 97.)

Projekti saatetaan loppuun onnistuneesti, mutta projektin tuotteella ei olekaan aitoa painoarvoa. Tämä tilanne syntyy helposti, jos projektiryhmä ei ole tehnyt projektin esitutkimusta kunnolla. Tällainen tilanne voi olla esimerkiksi uuden ohjelmiston tai järjestelmän käyttöönotto, josta ei lopulta ole todellista hyötyä yritykselle. (Cornford-Smithson 2006, 78 - 97.)

Tekniset ongelmat voivat luoda riskin etenkin teknologiaan painottuvissa projekteissa. Tekniikan pettäminen voi aiheuttaa useiden päivien viivästymisiä tai pahimmillaan tiedon tuhoutumisen. Siksi onkin tärkeää varmistaa tekniikan toiminta ja varmuuskopioinnin toimintavarmuus. Teknisiltä ongelmilta suojautuminen on yksinkertainen toimenpide, joka kuitenkin jää taka-alalle liian usein. (Cornford-Smithson 2006, 78 - 97.)

### 3.4 Suunnitteluvaihe (planning)

Käynnistysvaihetta seuraa suunnitteluvaihe. Projektisuunnitelma tehdään lähes poikkeuksetta projektin alkuvaiheessa, koska suunnitelma toimii projektin viitekehystenä koko projektin elinkaaren ajan.

Projektisuunnitelman lähtökohtana on yksinkertainen tehtävämäärittely. Tehtävämäärittely tehdään erilaisten rajausten kautta. Tehtävämäärittelyssä rajataan projektiin otettavat ja pois jäävät tehtävät. Projektin rajausvaiheessa katsotaan mitä erilaisia vaiheita tarvitaan lopputuloksen aikaansaamiseksi. Mikäli osa-alueita rajataan projektista pois aikataulustus- tai kustannussyistä, se on ilmoitettava selkeästi rajausta tehtäessä. (Projektisuunnitelma n.d.)

Suunnitteluvaiheen aikana projekti jaotellaan pienempiin osiin, joita kutsutaan tehtäviksi (tasks). Suunnitteluvaiheen aikaansaannoksena on projektisuunnitelma, joka toimii ohjaavana elementtinä projektin suoritusvaiheen ajan. Projektisuunnitelman teossa käytetään yleisinä työkaluina projektisuunnitteluohjelmistoja. (Virtanen 2000.)

Projektisuunnitelman teossa on aina pidettävä mielessä tehtävän työn laajuus. Pienille projekteille ei laaja projektisuunnitelma ole välttämättä tarpeellinen. Pienellä projektilla tarkoitetaan projektia, jonka kesto on vain muutamia päiviä, tai maksimissaan kaksi viikkoa. (Tampereen yliopiston verkkosivut.)

Projektisuunnitelman rajauksia tehdessä on projektitiimillä oltava tieto käytettävästä teknologiasta, mitä projektin läpivieminen vaatii. On olennaista hahmotella projektin eri tehtäväalueet ja niiden sidonnaisuus toisiin tehtäviin. Karkea prosessi kuvaelma projektin edistymisestä: (Projektisuunnitelma n.d.)

- sisältösuunnittelu
- tekninen suunnittelu
- toteutus eli vaiheet
- tehtäväkokonaisuudet
- käyttöönotto.

Projektisuunnitelman tehtävä on muodostaa lista projektin tehtävien ja virstanpylväiden suorituspäivämääristä. Tehtäville on määritettävä myös tehtävistä odotettavat tulokset. (Cornford-Smithson 2006, 78 - 97.)

Projektisuunnitelman laatiminen rohkaisee projektiryhmää täydellisyyden tavoitteluun projektissa. Suunnitelman avulla projektiryhmän on helppo nähdä milloin projekti ei ole aikataulussa ja milloin on aikaa rentoutua. (Cornford-Smithson 2006, 78 - 97.)

Projektisuunnitelmassa virstanpylväät sidotaan tiettyyn päivämäärään. Virs-

tanpylväät ovat projektin tärkeitä päivämääriä. Ne saattavat merkitä tiettyä tapahtumaa, kuten mahdollista tilannekatsausta asiakkaan kanssa, jolloin projektin on oltava tietyssä vaiheessa. Virstanpylväitä käytetään myös ilmaisemaan muita projektiin vaikuttavia päiviä, kuten esimerkiksi joulua. (Cornford-Smithson 2006, 78 - 97.)

Projektisuunnitelmaan tehdään vaiheistus. Tällä ei tarkoiteta pelkästään ajallista vaiheistusta, vaan vaiheistusta myös tehtäväkokonaisuuksiin. Koko projektin eteneminen ja toiminta perustuu voimassa olevaan projektisuunnitelmaan. Projektisuunnitelmaa muutetaan ja muokataan jatkuvasti projektin vaatimalla tavalla. Ellei projektisuunnitelmaa pidetä projektissa ajan tasalla joutuu projekti helposti tuuliajolle. (Projektisuunnitelma n.d.)

### 3.3.1 Projektin ositus, aikataulutusta ja vaiheistus

Projektin aikataulutusta on osa projektisuunnitelman tekoa. On mahdotonta suunnitella yleiskäypää aikataulutustyyliä, joka sopisi jokaiseen projektiin. Siksi jokaiseen projektiin on määritettävä oma ainutlaatuinen aikataulunsa. Projektin aikataulutusta voidaan karkeasti jakaa seuraaviin osiin: (Projektisuunnitelma n.d.)

- tehtävät, päävaiheet ja ensimmäisen vaiheen tarkennetut osatehtävät
- väliraporttien aiheet
- alustavan aikataulun eli tavoiteaikataulun määrittely.

Projektin aikataulutusta tehtäessä aluksi selvitetään projektin välitavoitteet ja vaiheet, sekä mitä eri tehtäviä vaiheisiin sisältyy. Projektin tehtäväkokonaisuuksia määriteltäessä tutkitaan, miten eri tehtäväalueet ovat sidoksissa toisiinsa ja mitkä ovat tärkeimmät riippuvuuspuisteet. (Projektisuunnitelma n.d.)

Yksi projektisuunnittelun vaaroista on tehdä liian tarkka suunnitelma. Projek-

tisuunnitelman teossa on hyvä pitää mielessä vanha metodi (KISS) *“keep it simple stupid”*. Cornfordin ja Smithsonin mukaan projektisuunnitelman koko ei saisi ylittää A4 kokoista arkkia. Suuri virhe pienen projektin suunnittelussa on käyttää liian laajaa työkalua sen suunnitteluun. Microsoft Project ohjelma ei esimerkiksi sovi välttämättä hyvin lyhytkestoisiin projekteihin. Suunnitelmasta tulee helposti liian yksityiskohtainen ja laaja projektin kokoon nähden.

Cornfordin ja Smithson (2006, 78 - 97) mukaan avaintekijöitä aikataulutuksessa ovat seuraavat asiat:

- Aikataulutus ei saa olla liian yksityiskohtainen.
- Projekti hajotetaan helpommin käsiteltäviin osiin.
- Suunnittele aikataulu joka varmistaa varhaisessa vaiheessa konkreettisia tuloksia.
- Varaa aikaa projektin loppuun tulosten tarkastamiseen ja dokumentointiin.
- Aikataulutusta ei saa tehdä liian tiukaksi.

Projektisuunnitelmaa tehtäessä työt on hyvä jakaa pieniin osiin siten, että yksi tehtävä ei veisi aikaa yli kahdesta kolmeen viikkoon. Projekti on helpommin hallittavissa sopivan kokoisissa osissa. Projektin jakaminen sopivan-kokoisiin osiin voi myös vaikuttaa projektiryhmän motivaatioon tehdä työtä. Liian suuret tehtäväkokonaisuudet vievät paljon aikaa ja suoranaista tulosta ei nähdä tarpeeksi nopeasti. (Cornford - Smithson 2006, 78 - 97.)

Yksi tärkeä tekijä projektisuunnittelussa on varata tarpeeksi aikaa projektin loppuvaiheeseen. Loppuvaiheessa on jäljellä useita raportointi- ja yhteenvetovaiheita, jotka vievät aikaa projektista. Pelkästään perustehtävien kuten tulostamisen, arkistoinnin ja tarkistamisen, läpivienti vie aikaa yllättävän paljon. (Cornford - Smithson 2006, 78 - 97.)

Projektinaikataulutuksessa on olennaista huomioida myös tehtävät, jotka eivät suoranaisesti liity projektiin mutta jotka ovat jatkuvasti projektiryhmän taakkana. Pelkästään sähköpostiviestien, ongelmien, matkustamisen, tapaa- misten ja muiden perustehtävien viemä aika saattaa olla useita työpäiviä. Työaikaa vie myös mahdollinen odottelu riippuen projektitehtävien luonteesta. Liian löysä aikataulutusta puolestaan aiheuttaa resurssien hukkaan menemistä ja taloudellisia tappioita. (Cornford - Smithson 2006, 78 - 97.)

Projektin luonteesta riippuen aikataulutuksessa on otettava huomioon myös työt, jotka pohjustavat projektia. Esimerkiksi tyytyväisyyskyselyissä on usein paljon haastatteluja ja kyselyitä, jotka suoritetaan osana projektin aloitusta. Näihin töihin kuuluva aika on myös otettava huomioon aikataulutusta tehtäessä, vaikka kyseiset tehtävät eivät ole helposti käsitettävissä työksi. (Cornford - Smithson 2006, 78 - 97.)

### 3.5 Suoritus- ja kontrollointivaihe (execution)

Suunnitteluvaihetta seuraa projektin suoritus- ja kontrollointivaihe. Suoritusvaiheessa toimitaan projektisuunnitelman pohjalta, eli projektin tehtävät suoritetaan suunnitelman mukaisesti. Suoritusvaiheen alkuvaiheessa projektiryhmällä on jaoteltu roolit ja tehtävät. Nämä määrittävät määrittely- ja suunnitteluvaiheessa. (Project perfect n.d.)

Tärkeä asia suoritusvaiheessa on saada projektin tehtävät suoritettua aikataulussa. Projektisuunnitelmaan merkitään yleensä virstanpylväitä (milestones), jotka helpottavat suoritusvaiheen seurantaan. Suoritusvaiheen lopputuloksena on yleensä valmis tuote, prototyyppi, sovellus tai tutkimus. (Virtanen 2000.)

Suoritusvaiheessa projektisuunnitelmaa käytetään suorituksen seurantaan ja aikataulutukseen. Seurannan tärkeyttä ei tule vähätellä. Suoritusvaiheen ominaispiirteisiin kuuluukin projektisuunnitelman päivittäminen nykyistä tilaa vastaavaksi. Seurannan avulla on mahdollista luoda projektista myös välira-

portteja, jotka ovat usein tärkeitä asiakkaille ja osakkeenomistajille. Raportointi projektin aikana helpottaa huomattavasti projektin seurantaan. Raportointia käytetään niin sisäiseen kuin ulkoiseenkin seurantaan. (Arkansas DoIS execution phase 2006.)

On tärkeää, että projektin suoritus nojautuu täydellisesti projektisuunnitelmaan. Suunniteltuvaiheessa tehtyjä virheitä voidaan korjata ja aikataulutusta muokata tarvittaessa, mutta suunnitelman noudattaminen on tärkeää aikataulun pitämisessä. Mikäli suunnitelmaa muokataan ja muutetaan suorituksen mukaan, on aika miettiä, mikä suunnittelussa on mennyt pieleen ja onko mahdolliset riskit kartoitettu hallitusti. (Arkansas DoIS execution phase 2006.)

Suoritusvaiheessa pyritään hallitsemaan ja seuraamaan riskitilanteita. Riskien tunnistaminen hyvissä ajoin antaa projektiryhmälle enemmän pelivaraa jo toteutuneen riskin vahingon minimointiin. Riskinhallinta dokumenttia, joka luotiin projektin käynnistysvaiheessa, käytetään riskien tarkkailun perustana. Riskihallinnan dokumentin tulisi sisältää valmiiksi suunnitellut toimintamallit riskien vahingon minimointiin. (Arkansas DoIS execution phase 2006.)

### 3.6 Lopetusvaihe (closure)

Suoritusvaihetta seuraa projektin lopetusvaihe. Lopetusvaiheessa tehdään yhteenveto projektista. Projektipäällikön tehtävänä on tarkistaa, että projekti on tuotu kunnialla loppuun ja projektin tuotos on halutunlaatuinen. Lopetusvaiheessa saadaan viimeinen mielipide asiakkaalta tuotoksen laadusta ja tuloksesta. (Virtanen 2000.)

Lopetusvaiheen tärkeimpänä tavoitteena on projektin päättäminen. Projekti ei pääty itsestään, vaan se on päätettävä. Suoritetaan arviointi syntyneistä tuloksista, sekä kielteisistä että myönteisistä. Välitetään tietoa välittömille sidosryhmille projektiarvioinneista saaduista tuloksista. (Virtanen 2000.)



Petri Virtasen mukaan projektin päättämisen voi jakaa neljään eri kategoriaan. Projekti voi yksinkertaisesti ”kuolla”. Projektin tarpeellisuus katoaa syystä tai toisesta. Syyt riippuvat projektin osalta projektin tyypistä tai siitä, että projektin tarve on hävinnyt esimerkiksi epärealististen tavoitteiden takia. (Virtanen 2000.)

Toinen tapa päättää projekti on sen korvaaminen jollakin muulla toiminnalla. Usein tämä tapahtuu, kun huomataan, että projekti on hyödyllinen esimerkiksi tehostamaan yrityksen linjatyötä. Tästä syystä se muutetaan osaksi emo-organisaation toimintaa. (Virtanen 2000.)

Kolmantena tapana projekti integroidaan toiseen projektiin, jonka tuloksena käytännössä syntyy uusi projekti. Projekti voi alkuaankin olla isomman kokonaisuuden aliprojekti. (Virtanen 2000.)

Neljäs tapa projektin päättämiseen on näännyttää projekti nälkään. Tämä syntyy usein rahoittajan rohkeuden puutteesta. Asiakas käytännössä ei uskalla suoraan esittää projektinlopettamista vaan se lopetetaan pala palalta. (Virtanen 2000.)

Miten tahansa projekti päätetäänkin, on siitä saatu tietoa erittäin arvokasta seuraavia projekteja suunniteltaessa. Projektin päättämistä ei kannata aliarvioida, sillä se on erittäin tärkeä osa projektin kokonaisuutta. Projektia suunniteltaessa on mietittävä, miten projekti tullaan päättämään. Varmistetaan, että kaikki haluttu informaatio projektista on tallessa ennen sen hautaamista. Hyvän projektinhallintaohjelmiston ominaisuuksista onkin löydyttävä kattava dokumentointi järjestelmä, tai minimissään pieni arkistointiominaisuus.

## 4 WEB-SOVELLUSTEN TOIMINTA JA RISKIT

Osiossa käsitellään web-sovelluksen alustavaatimuksia ja eri alustaratkaisujen rakennetta. Projektinhallintasovelluksen monimutkainen arkkitehtuuri asettaa korkeat vaatimukset yrityksen sisäverkolle (intranet). Lisäksi käsitellään myös lyhyesti web-sovellusten yleisimpiä riskejä ja sitä, kuinka niitä voidaan välttää.

### 4.1 Yleistä

WWW on kehittynyt informaation lähteestä monimutkaisten sovellusten toimintaympäristöksi. Uusia teknologioita kehitetään jatkuvasti, mikä tekee web-sovelluksen määrittelyn mahdolliseksi. Web-teknologioiden kehitystä vauhdittaa asiakkaalle suunnattujen palveluiden monimuotoisuus ja yritysten globalisoituminen. Tämä luo tarpeen saada web-ympäristöön mahdollisimman työpöytämäisen sovelluksen, jossa tieto on jatkuvasti ajan tasalla.

Alustariippumattomuus on erityisen tärkeää suunniteltaessa web-sovellusta. Hyvin suunniteltu ja teknologia-riippumaton sovellusarkkitehtuuri takaa kontrolloidun ympäristön eri teknologioiden yhdistämiseen. Huomioon otettavaa on myös se, että iso osa teknologioista on kieliriippuvaista. Eri moduulit web-sovelluksessa on ohjelmoitu eri ohjelmointikielillä. HTML, PHP, Java, CGI, C/C++, C#, Java ja SQL ovat vain pieni osa olemassa olevista tekniikoista. Tämä ei vain kerro vain sovellusalustojen laajasta valikoimasta, vaan myös teknologiariippumattoman web-sovelluksen konseptista. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

### 4.2 Web-palvelin

Web-palvelin muodostuu erilaisten sovellusmoduulien muodostamasta ryhmästä. Nämä sovellusmoduulit on asennettu verkkopalvelimeksi tarkoitettulle

palvelintietokoneelle, joka kommunikoi verkon kautta ulkomaailmaan. Web-palvelin käsittelee HTTP-pyyntöjä, välittää pyynnöt seuraavaksi web-sovellukselle ja palauttaa sovelluksen antaman tuloksen asiakkaalle. Asiakkaana toimii lähes poikkeuksetta web-selain. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

HTTP-vastaus sisältää usein HTML- tai XML-tyyppistä sisältöä. Poikkeuksellisesti palvelin voi palauttaa pyynnöstä riippuen esimerkiksi kuvan, tiedoston, tai jonkin muun tyyppisen tiedoston, joka määritellään tiedoston MIME-tyypissä. Web-palvelin sisältää ylläpitäjän kannalta myös muita elintärkeitä palveluita, kuten tapahtumalokin tai autentikaation suojaetuille alueille. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Web-palvelin tarvitsee palvelinkäyttöjärjestelmän, -ohjelmiston tai moduulin toimiakseen. Palvelinsovellukset luovat ympäristön web-sovellusten suorittamiselle ja sisältävät sovellusten tarvitsemat moduulit.

Palvelinsovelluksiin on mahdollista myös asentaa tarvitsemiansa moduuleita jälkepäin. Niitä voivat olla esimerkiksi eri ohjelmointikielten vaatimat moduulit. Kukin kieli tarvitsee oman moduulin toimiakseen. Moduuli voi toimia samalla myös sovelluskehysenä. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Netcraft verkkosivuston tilasto joulukuussa 2007 käytetyimmistä palvelinalustoista: (Netcraft Web Server Survey 2007.)

- Apache, Apache (76 591 442 sivustoa),
- Microsoft, IIS (55 502 886 sivustoa),
- Google, GFE (8 555 391 sivustoa).

### 4.3 Web-sovellusten alustat

Web-sovelluksen perusvaatimuksena on palvelinalusta ja sille asennetut tarvittavat moduulit. Jokainen web-sovellus tarvitsee melkein poikkeuksetta ympärilleen ohjelmistokehysten (framework). Ohjelmistokehys muodostaa rungon päällerakennettavaan sovellukseen. Kehykset ovat käytössä myös työpöytäohjelmistoissa. Kehys tarjoaa valmiiksi rakennettuja kokonaisuuksia, joita ei tarvitse ohjelmoida uudelleen. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Termi alusta pitää sisällään erilaisia muodostelmia sovellusmoduuleista, jotka on konfiguroitu toimimaan yhdessä useiden web-sovellusten alustana. Eri alustamoduulit palvelevat eri tarkoitusta ja lisäävät eri tekniikoita käyttöön. Alustat voidaan luokitella eri kerroksiin, eli web-selain kuuluu osana asiakas-kerrokseen ja HTTP-palvelin toimii osana web-kerrosta. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Web-arkkitehtuurissa on yleistä monen sovelluksen suhde toisiinsa. Web-sovelluksissa on usein käytössä useita eri kehyksiä ja arkkitehtuureita. Tämä tekee niistä ainutlaatuisia työpöytäsovelluksiin nähden. Sovellusta voidaan laajentaa käyttämällä toista arkkitehtuuria toisessa alustassa. Monimuotoisuus mahdollistaa sovelluksissa melkein rajoittamattoman toimintaympäristön. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Jokainen sovellus, riippumatta siitä kuinka yksinkertainen tai monimutkainen se on, tarvitsee jonkin tietyn moduulin toimiakseen. Sovellus toimii moduulin (module) päällä käyttäen sen tarjoamaa toiminallisuutta. Moni sovellus toimii yhdellä alustalla, mutta haluttaessa alustoja voidaan ottaa käyttöön useita. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Sovelluserroksen perustana toimii fyysinen kerros, joka pitää sisällään sovelluserroksen edellyttävät laitteet. Fyysiseen kerrokseen kuuluu palvelinlait-

teisto ja verkon infrastruktuuri. Verkon infrastruktuuri pitää sisällään erilaisia verkon toimintaan vaikuttavia laitteita, kuten palomuurit, kytkimet, reitittimet jne. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Web-sovellukset toimivat useilla arkkitehtuureilla toteutetuilla fyysisillä alustoilla. Sovellukset eivät yleisesti enää ole alustariippuvaisia, vaan ennemminkin riippuvaisia oikeista moduuleista. Esim. Java-sovellukset tarvitsevat Java-ympäristön palvelimelta toimiakseen. Web-sovelluksien suurin poikkeavuus työpöytäsovelluksiin nähden on juuri riippumattomuus palvelinkäyttöjärjestelmästä, mutta ne ovat riippuvaisia moduuleista. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

On olemassa arkkitehtuureja alustalle ja arkkitehtuureja sovellukselle. Kukin arkkitehtuuri tarjoaa omat ominaisuutensa, joita sovellus voi käyttää hyväksi. Eri arkkitehtuurit myös rajoittavat sovelluksen toimintaa. Alustan eri arkkitehtuurit ja komponentit toimivat yhdessä ja tarjoavat sovellukselle ajon aikaisen tuen ja toimintalogiikan. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Jotta web-sovellus olisi käytettävissä selaimella, sen alla on myös verkkoinfrastruktuuri. Tietoverkot koostuvat erilaisista verkkolaitteista, jotka kommunikoivat keskenään Mac-osoitteiden avulla. Web-sovelluksen alla toimiva serveri ja tietoverkon infrastruktuurin taso sanelevat sovelluksen näkyvyysalueen. Näkyvyysalueella tarkoitetaan sitä, onko sovellus saatavissa vain sisäverkossa vai onko se näkyvissä koko maailmalle. Verkon infrastruktuuri saattaa olla erittäin monimutkainen ja koostua lukemattomista laitteista. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

#### 4.4 Web-tekniikat

Web-tekniikat ovat osa web-alustan eri kerroksia ja toimintoja. Kukin web-sovellus vaatii taustalleen toimivia web-tekniikoita sovelluksen suorittamiseen. Erityisesti laajat sovellukset vaativat useita eri web-tekniikoita toimiak-

seen, kuten esimerkiksi projektinhallintasovellus. Laajat web-sovellukset tarvitsevat myös lähes poikkeuksetta tuen tietokantamoottorille, tai vaihtoehtoisesti korvaavan tavan tallentaa tietoa. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Web-sovelluksen eri teknologiat vaativat web-alustalta erilaista tukea. Kuvitellaan sovellus, joka käyttää Ajax, Js, PHP, C#, HTML, SQL, CSS ja XML teknologioita hyväkseen. Kyseisen sovelluksen vaatima alustatuki eri teknologioille on vaativa. Kyseinen sovellus olisi mahdollista pystyttää toimimaan joko Linux- tai Windows-server-käyttöjärjestelmän päälle. Sovellus vaatisi toimiakseen ainakin seuraavat asiat: (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

- Palvelinraudan, tietokone, joka toimii palvelimena. Myös perus verkkoinfrastruktuuri on oltava olemassa, mikäli sovellusta halutaan ajaa palvelimen ulkopuolelta.
- Käyttöjärjestelmän, kuten Windows Server, Linux, Solaris, Mac-server, tai muu palvelinkäyttöjärjestelmä
- .NET-kehiksen C#-kielelle Windows-ympäristössä, tai Mono laajennusmoduulin Linux-, Solaris-, Mac OS X- ja UNIX-ympäristöissä
- PHP-palvelinmoduulin asennettuna palvelimen käyttöjärjestelmään
- Windows-ympäristössä IIS:n (Internet Information Service) tai Linux-ympäristössä Apachen ja Mono Netin. Kyseiset teknologiat ovat palvelinsovelluksia, eli toiselta nimeltään HTTP-palvelinohjelmia.
- SQL-kielen käyttö vaatii palvelinkäyttöjärjestelmältä tietokantamoottorin, kuten esimerkiksi MySQL, MS SQL, PostgreSQL tai Oracle.

- Ajax, Js, HTML, CSS ja XML–teknologioiden käyttö ei vaadi palvelinlustralta erityistä ohjelmistoa. Kukin teknologia vaatii kuitenkin asiakaspuolen selaimelta tuen kyseisille teknologioille.

Web-sovelluksen monimutkaisuus johtuu juuri sen tavasta käyttää useita eri tekniikoita hyväkseen samaan aikaan. Web-sovellustekniikat kehittyvät nopeasti ja muuttavat muotoaan. Tämä aiheuttaa paineita sovellusten kehittäjille, joiden on otettava huomioon useita asioita kerralla kehittäessään sovellusta. (Jablonski, Petrov, Meiler, Mayer 2004, 3 - 37.)

Web 2.0:n myötä erilaiset asiakaspään scriptikielet ovat nousseet sovelluskehittäjien suosioon. Kokonaisten sivustojen HTTP-kutsut on korvattu AJAX-ohjelmointitekniilla (Asynchronous JavaScript And XML). AJAX:n avulla HTTP-kutsu voidaan tehdä vain tietylle osalle sovellusta, eikä koko sivustoa tarvitse ladata uudelleen. Asiakaspuolen scriptit tekevät sovelluksen toiminnallisuudesta enemmän työpöytäsovelluksen kaltaisia. (Asp.net Ajax n.d.)

#### 4.5 Web-ympäristön riskit

Web-sovelluksen toimintaympäristö ei ole riskitön. Alustan ja ympäristön luomat riskit poikkeavat työpöytäsovellusten riskeistä. Siksi web-sovelluksen riskihallinta vaatii suunnitelman riskien hallintaan. Suunnitelman on hyvä sisältää lista toimintamalleista, joilla riskejä torjutaan. Neljä pääriskiä web-sovelluksissa ovat (WASC Threat classification 2004.):

- Käyttäjän autentikaation epäonnistuminen, joka aiheuttaa luvattomien persoonien pääsyn sovellukseen ja sen sisältämään tietoon. Tämä on usein mahdollista suoraan web-käyttöliittymän kautta.
- Virheet sovelluksen koodissa, jolloin aiheuttavat pääsyn sovelluksen lähdekoodiin.

- Suojaamattomat yhteydet, tieto on kaapattavissa HTTP-paketeista suoraan.
- Sovelluksen, tai verkkoinfrastruktuurin huonosti suunniteltu tietoturva.

Web-ympäristöön liittyy myös paljon riskejä tietoturvan ulkopuolelta. Riskit saattavat johtua verkkoinfrastruktuurin heikosta rakenteesta, palvelimen riittämättömyydestä, tiedon varmuuskopioinnin puutteesta tai lukemattomista muista syistä.

Alla luetellaan jokaiseen web-sovellukseen sisältyvät riskit. Alla olevat menetelmät ovat yleisesti käytettyjä web-sovellusten kaappaamiseen ja tie arkaluontoisen tiedon joutumiseen väärin käsiin. Jokainen alla oleva menetelmä voidaan luonnehtia vakavaksi. Yleisimpiä menetelmiä sovellusten kaappaamiseen ovat brute force -menetelmä ja SQL-injektio.

#### 4.5.1 Brute force

Brute force on menetelmä, jossa automaattinen prosessi yrittää arvata käyttäjätunnusta ja salasanaa. Useissa järjestelmissä salasanaa ei ole kryptattu tarpeeksi hyvin ja se on murrettavissa brute force menetelmällä. Brute force menetelmän avulla voidaan myös murtaa esimerkiksi käyttäjän luottokorttinumero. (WASC Threat classification 2004.)

Brute force menetelmiä on kahdelaisia normaali menetelmä ja käännteinen menetelmä. Normaalisissa menetelmässä botti yrittää ratkaista yhden käyttäjätunnuksen ja etsiä sille yhtä salasanaa. Käännteisessä menetelmässä botti yrittää arvailla usealle käyttäjätunnukselle useaa salasanaa. (WASC Threat classification 2004.)



#### 4.5.2 Puutteellinen autentikaatio

Puutteellisella autentikaatiolla tarkoitetaan pääsyä suojatulle alueelle ilman asianmukaista autentikointia. Käytännössä käyttäjä pääsee suojattuun materiaaliin käsiksi ilman tunnistusta. Web-pohjaiset hallintatyökalut ovat yleinen esimerkki sovelluksista, jotka kärsivät puutteellisesta autentikaatiosta. (WASC Threat classification 2004.)

Yleisesti autentikaation puuttuminen tulee esille sovelluksessa, jossa sisältö on suojattu piilottamalla se käyttäjältä. Yleinen tilanne suojaamattomassa sisällössä on kirjoittaa URL:n viittaus `"/admin"` virtuaaliseen kansioon. (WASC Threat classification 2004.)

#### 4.5.3 Salasanan palautustoiminto

Salasanan palautukseen liittyvä heikko käyttäjän validointi aiheuttaa uuden salasanan joutumisen väärin käsiin. Sovellus voi palauttaa käyttäjän salasanan virheelliseen sähköpostiosoitteeseen tai tulostaa uuden salasanan hyökkääjän ruudulle validoimatta, kuka uuden salasanan pyytäjä on. (WASC Threat classification 2004.)

Sovelluksessa saattaa olla käyttäjän rekisteröinti vaiheessa asettama kysymys, tai vinkki salasanan palautukseen. Mikäli hyökkääjällä on tieto palautettavan salasanan käyttäjästä, hän voi tietää esimerkiksi käyttäjän sosiaaliturvatunnuksen. Salasanan palautustoimintoa voidaan käyttää myös brute force -menetelmässä hyväksi. Brute force -menetelmä voi esimerkiksi käyttää lemmikkien nimi tietokantaa käyttäjän lemmikin nimen arvaamiseen. (WASC Threat classification 2004.)

#### 4.5.4 Istunnon kaappaaminen

Istunnon kaappaamisessa hyökkääjä kirjautuu käyttäjän identiteetillä sovel-

lukseen. Sovellus tunnistaa tämän jälkeen hyökkääjän esimerkiksi käyttäjänä ja antaa hyökkääjälle oikeudet tehdä käyttäjäroolille sallittuja toimintoja. (WASC Threat classification 2004.)

Istunto voidaan myös kaapata saamalla selville käyttäjän istuntotunniste. Tunniste on aina yksilöllinen päällekkäisten istuntojen välttämiseksi. Varastetun istuntotunnisteen avulla käyttäjän on mahdollista esiintyä toisen käyttäjän identiteetillä. (WASC Threat classification 2004.)

#### 4.5.5 Asiakaspään hyökkäykset

Asiakaspään hyökkäykset aiheutuvat usein käyttäjän koneella olevista vakoi-  
luohjelmista, jotka kaappaavat käyttäjän luoman istunnon palvelimelle. Käyt-  
täjän kirjautuessa palveluun botti ohjaa kirjautumistiedot omalle palvelimel-  
leen asiakkaan huomaamatta. Kyseinen menetelmä on erittäin vaarallinen ja  
sen avulla voidaan kaapata esimerkiksi asiakkaan pankkiyhteys. (WASC  
Threat classification 2004.)

Vakoilusovellukset käyttävät myös usein tekniikkaa, jossa käyttäjä ohjataan  
väärään sisältöön hänen tietämättään. Tässä käytetään usein hyväksi HTML-  
kehyksiä joihin ladataan HTML-sisältö erilliseltä sivulta. Käyttäjän näkökul-  
masta hän selaa oikeaa sivustoa, mutta kirjautuessaan luovuttaakin tunnistau-  
tukseen tarvittavat elementit kolmannelle osapuolelle. (WASC Threat classi-  
fication 2004.)

#### 4.5.6 SQL-injektio

SQL-injektio-hyökkäys on yleisesti käytetty tapa korruptoida ja saada arka-  
luontoista tietoa. SQL-injektio toteutetaan syöttämällä tietokantakyselyyn ky-  
selyiden osia, jotka vääristävät kyselyt erilaisiksi. Väärällä kyselyllä hyökkää-  
jän on mahdollista saada tietokannasta ulos arkaluontoista tietoa, tai voidaan  
kirjautua väärällä tunnisteella sisään. (WASC Threat classification 2004.)

SQL-injektio voidaan estää rakentamalla sovellus siten, että sen kautta ei voida syöttää väärää tietoa. Usein injektio toteutetaan web-lomakkeen kenttään syötetyistä virheellisistä tiedoista. LDAP-injektio toimii samoilla metodeilla kun SQL-injektio. (WASC Threat classification 2004.)

## 5 KÄYTETTÄVYYS

Osiossa käsitellään web-sovellusten käytettävyyttä, erityyppisiä käytettävyyden arviointimetoodeja sekä sitä miksi tietty metodi on valittu tukemaan suoritettavaa ohjelmistovertailua.

Lisäksi tutkitaan sovellusten käytettävyyttä ja sitä käytettävyydestauksessa on otettava huomioon. Mietitään myös, millainen on web-sovelluksen yleinen käyttöliittymärakenne ja mitä käyttöliittymän suunnittelussa tulisi ottaa huomioon hyvän käytettävyyden saavuttamiseksi.

Osion loppuosassa on tietoa tutkimuksessa käytettävästä PSSUQ-menetelmästä. PSSUQ-menetelmää käytetään sovellusten käyttöliittymän käytettävyyden vertailuun ja testaamiseen.

### 5.1 Käytettävyys web-sovelluksessa

Sovelluksen käytettävyyteen on syytä kiinnittää huomiota, sillä web-sovellukset asettavat käytettävyydelle erilaiset mittasuhteet kuin työpöytäsovellukset. Hyvä käytettävyys sovelluksessa vaikuttaa suoraan asiakastyytyväisyyteen siten sovelluksen markkinoihin. (Brinck ym. 2002, 2-25.)

Mikä sitten on käytettävyydeltään hyvä web-sivusto? Suoranaista kaavaa hyvälle käytettävyydelle ei voi määrittää. Käytettävyyteen vaikuttaa osaltaan konteksti ja osaltaan myös sivustoa käyttävät ihmiset. (Brinck ym. 2002, 2-25.)

Käytettävyys voidaan määritellä seuraavalla yksikertaisella metodilla: millä osaamistasolla oleva käyttäjä voi suorittaa vaadittavia tehtäviä sovelluksella. Käytettävyysuunnittelun määränpäiksi voidaan määritellä seuraavat asiat: (Brinck ym. 2002, 2-25.)

- Sovellus toimii sille tarkoitetulla tavalla, eli sovelluksessa ei saa olla virheitä eikä epäloogisuuksia toimintamallissa.

- Käytettävyys tehokkuus. Sovelluksen käytön ja toiminnan on oltava nopeaa. Käytännöntasolla proseduurit, jotka ovat nopeampia käyttää, ovat myös laadukkaampia.
- Sovelluksen käyttö on helppo oppia. Käyttäjän on opittava sovelluksen käyttö kohtuullisessa ajassa. Käytännössä mitä vähemmän askelia proseduurin ajaminen vaatii, sitä parempi on sovelluksen käytettävyys.
- Sovellus on helposti muistettava. Käyttäjän on helppo muistaa sovelluksen toimintaperiaate ja proseduurien vaatimat askeleet sovellusta käytettäessä.

Web-sovellus koostuu kohtuullisen yksinkertaisesta rakenteesta. Käyttöliittymä rakentuu linkeistä, napeista, valikoista, lomakkeista ja yksinkertaisesta grafiikasta. Yksinkertaisesta rakenteesta huolimatta web-sovellusten käyttö tuottaa vaikeuksia useille ihmisille. Web-sovelluksille on yleistä rakenne, jossa päänavigointivalikko on sivun ylälaidassa ja ns. matalamman tason dynaaminen valikko sivun vasemmassa tai oikeassa laidassa. Kuitenkin jo hieman yleisestä asettelusta poikkeava sivusto aiheuttaa käyttäjille vaikeuksia löytää etsimäänsä. (Brinck ym. 2002, 2-25.)

Selaimen välimuisti saattaa heikentää sovelluksen käytettävyyttä. Välimuisti parantaa suorituskykyä ja kuormittaa palvelinta vähemmän, mutta aiheuttaa sovelluksissa välillä odottamatonta käyttäytymistä. Käyttäjän palatessa sivustolle lyhyen ajan jälkeen sivusto palautetaan selaimen välimuistista. Uusi informaatio jää käyttäjän ulottumattomiin, kun vanha sivu palautetaan välimuistista. Välimuistia voidaan kuitenkin hallita osaksi myös palvelinpäästä, mikä on otettava ottaa huomioon sovellusta tehtäessä. (Brinck ym. 2002, 2-25.)

Lopullisia sovelluksen käyttäjiä ei ole viisasta käyttää sovelluksen testaajina. Sovelluksen testaaminen tulisikin suorittaa ennen julkaisua. Huonot päätökset käytettävyiden suhteen on korjattava välittömästi suunnittelun ja ohjelmoin-

nin aikana. Käytettävyyssuunnittelun kohde on aina oltava käyttäjä, sillä sovellus on arvoton ilman käytettävyyttä. (Brinck ym. 2002, 2-25.)

## 5.2 Käytettävyyden mittaaminen

Käytettävyyden mittaamiseen on kehitetty paljon erilaisia asteikkoja ja mittareita. Näiden avulla saadaan tietoa tutkittavan tuotteen käytettävyydestä. Tutkimuksessa mitataan tyytyväisyyttä sovelluksista ja saadaan tietoa sovelluksen käytettävyydestä. Parhaiten ohjelmistojen käytettävyyden mittaamiseen soveltuvat kyselylomake tyyppiset menetelmät. Software Usability Measurement Inventory (SUMI), The Questionnaire for User Interaction Satisfaction (QUIS) ja Post-Study System Usability Questionnaire (PSSUQ) on suunniteltu mittaamaan käytettävyyttä.

## 5.3 Sovellettava menetelmä

Sovellusten testaamiseen käytetään PSSUQ-menetelmää. Lewis (1995) esittää sarjan kyselylomakkeita käytettävyyden arvioimisen eri tilanteita varten. Yksi näistä (After Scenario Questionnaire ASQ) sopii käytettäväksi välittömästi kunkin testitehtävän jälkeen, toinen (Post Study System usability questionnaire PSSUQ) koko käytettävyydestin päätyttyä ja kolmas (Computer System Usability Questionnaire CSUQ) soveltuu kenttätyöhön. Kaikki ovat IBM:n kehittämiä. (Keinonen. n.d.)

- PSSUQ ja CSUQ käyttävät lähes samankaltaisia menetelmiä. Kysymykset käsittelevät kolmea asiaryhmää: kelpoisuus, tietojen laadukkuus ja käyttöliittymän laadukkuus.
- Kelpoisuus (system usefulness) -otsakkeen alla kysytään käyttäjän arvioita käytön helppoudesta, opittavuudesta, nopeudesta ja tehosta tehtävien suorittamisessa sekä subjektiivista tuntemusta.

- Tietojen laadukkuus (information quality) -kohdassa pyydetään käyttäjää arvioimaan järjestelmän virheilmoituksia ja virheistä toipumista, tietojen selkeyttä ja ymmärrettävyyttä.

Käyttöliittymän laadukkuus (interface quality) kartoittaa käyttäjän tyytyväisyyttä näppäimistöön, hiireen, kuvaruutuun jne. Näillä asteikoilla mitataan lähinnä käyttäjän tunneperäisiä asenteita. (Keinonen. n.d.)

Parhaiten tutkimuksen käytettävyyssvertailua tukee PSSUQ-menetelmä. Tämä menetelmä on helposti toteuttavissa ja antaa silti hyvin paljon tietoa ohjelmiston käytettävyyteen. Tutkimuksessa ei oteta kantaa siihen miten maksullinen vastaan maksuttomat ohjelmistot tulisi pisteyttää.

## 6 TESTATTAVAT PROJEKTIHALLINTASOVELLUKSET

Seuraavassa osiossa käsitellään tilannetta, jossa oltiin ja joka tullaan toteuttamaan. Mitkä ohjelmistot selvisivät rajausten sisään ja mitkä rajattiin pois tutkimuksesta? Pois rajattuja ohjelmistoja ei mainita seuraavassa luvussa.

### 6.1 Alustavat toimenpiteet

Tavoitteena on päästä tilanteeseen, jossa voimme itse nähdä, mihin kukin sovellus pystyy. Useasti sovelluksia markkinoitaessa luvataan paljon niiden ominaisuuksien suhteen, kuten tutkittujen sovellusten osalta voitiin myös todeta. Käytettävyys nousi suureen arvoon sovelluksia tutkittaessa. Usein projektinhallintasovellukset toimivat samankaltaisen toimintamallin avulla, joten ulkoasu ja käytettävyys muodostuivat ratkaisevaksi tekijäksi sovelluksia valittaessa ja vertailtaessa.

Tutkimuksen ensimmäisen vaiheen aikana etsittiin web-pohjaisia projektinhallintaohjelmistoja ilman rajoituksia. Lopulliset rajoitukset muodostuivat osaksi sovellusten yleisten toimintaperiaatteiden perusteella.

Projektinhallintasovelluksia otettiin mukaan alustaviin ominaisuustutkimuksiin 35 kappaletta, joista valittiin parhaat vaihtoehdot testaukseen. Testattavien sovellusten tuli täyttää seuraavat kriteerit:

- Ohjelmiston on pystyttävä käsittelemään useampaa kuin yhtä projektia.
- Sovelluksen on toimittava web-ympäristössä.
- Ohjelmistokehityksen taustalla toimii aktiivinen yhteisö.
- Käyttönoton kustannusarvio, käyttönotosta ei kohtuuttomia kustannuksia asiakkaalle.
- Sovelluksen käyttöliittymä on selkeä.



- Sovellus on mahdollista asentaa paikallisesti, eli ei ohjelmistovuokrausta (Application Service Provider, ASP).

Tutkimukseen otettiin tarkasteluun myös laajempien sovellusten projektinhallinta ominaisuudet. Tästä syystä mukaan otettiin yksi CRM sekä ERP-sovelluskokonaisuus. Tutkittiin vain niiden tarjoamia projektinhallintaan liittyviä ominaisuuksia, kuinka hyvin projektinhallinta toimii suuren sovelluskokonaisuuden alaisuudessa sekä projektinhallintaominaisuuksia käytettävyyden näkökulmasta.

Tutkimuksessa mukana olleet projektinhallintasovellukset poikkeavat toisistaan pääasiassa ohjelmointikielen, alustavaatimusten ja toimintaperiaatteen eroavaisuuksilla. Testaukseen valitut sovellukset ovat:

1. Easy project.net
2. activeCollab
3. tinyErp
4. SugarCRM
5. trac
6. DotProject

Sovelluksen käytettävyydellä tarkoitetaan sitä, kuinka yksinkertaisesti sovelluksella on mahdollista toteuttaa toimintoja. Käytettävyyteen vaikuttaa myös sovelluksen graafinen ulkoasu, joka on osa sovelluksen typografiaa. Testauksessa otettiin huomioon käyttöliittymän ulkoasusta seuraavat asiat:

- yhtenäinen ulkonäkö,
- yhtenäinen suunnittelu
- tieto on selkeässä järjestyksessä
- navigointivalikoiden helppo saatavuus ja selkeys
- esteettinen miellyttävyys.

Sovellustestaus suoritettiin kevyellä toiminta- ja käytettävyydestillä. Testauksella pyrittiin etsimään sovelluksen käyttöön liittyvät ongelmat ja haasteet. Sovellustestauksessa rajattiin teknologian vaikutus pois testauksesta. Testauksessa ei määritelty käytettävyystavoitteita tarkasti sovelluksien erilaisuudesta johtuen. Sovellukset pyrittiin testaamaan niiden todellisessa toimintaympäristössä.

Testeissä keskityttiin projektiryhmän tärkeimpiin ominaisuus-tarpeisiin, kuten projektin luonti ja tehtävien luonti. Sovelluksista pyrittiin löytämään vakavia käytettävyysongelmia. Testauksessa on muistettava kuitenkin testauksen ydin: käyttäjän tehtävä ei ole käyttö, vaan tavoitteeseen pääsy.

Projektinhallintasovelluksia testattiin niiden omissa toimintaympäristöissä. Tämä tarkoittaa käytännössä testiympäristön pystyttämistä sovelluksen vaatimuksien mukaan. Sovellusten testaamiseen käytettiin yksinkertaista testaus-suunnitelmaa.

Testauksen tyyppiä voidaan kutsua systeemitestaukseksi. Systeemitestauksen päätavoitteena on testata, täyttääkö valmis sovellus määritellyt vaatimukset.

Testaussuunnitelma piti sisällään seuraavat testitapahtumat:

1. Sovellukseen luotiin järjestelmänvalvojan roolissa oleva käyttäjä ja työntekijän roolissa oleva käyttäjä, mikäli sovelluksen ominaisuudet sallivat käyttäjäroolitukset.
2. Sovellukseen luotiin asiakas.
3. Sovellukseen luotiin projekti.
4. Projektille luotiin tehtäviä, virstanpylväitä ja tehtäväryhmiä, mikäli sovelluksessa oli siihen mahdollisuus.
5. Sovelluksen muita ominaisuuksia ja niiden toimivuutta testattiin. Näitä ominaisuuksia olivat mm. kalenteri, Gantt-kaavio, tiedostojenhallinta ja sähköposti-muistutukset.

## 6.2 Easy Projects .NET

Easy Projects .NET (EP) on kaupallinen projektihallintasovellus. EP-sovellus on suunniteltu yksinkertaiseksi projektihallintasovellukseksi helpottamaan projektien hallintaa. EP:stä on tarjolla sekä asennettava paketti että ASP-periaatteella toimiva kokonaisuus. EP:n lisenssi hinnoitellaan aktiivisten käyttäjien lukumäärän mukaan. Sovellukseen on mahdollista saada lähdekoodi mukaan \$ 2500: lisähinnasta.

EP sisältää perus-ominaisuudet projektihallintaan ilman mitään erityisiä lisätyökaluja. Sovellus sisältää seuraavat ominaisuudet: hierarkkinen projektihallinta, resurssien ajanhallinta, ajankäytön arviointi-työkalu, ongelma seuranta, virstanpylväät projektille, sähköposti ilmoitukset, ilmoitustaulu, tiedostonhallinta, raportointityökalut, kalenteri ja monia muita projektihallintaa helpottavia ominaisuuksia ja työkaluja.

### 6.2.1 Easy Projects .NET -asennus ja käyttöönotto

EP:n arkkitehtuuri on Microsoft ASP.NET 2.0 ja sovelluksen kielenä on C#. Sovellus on rakennettu käyttäen kolmikerros-arkkitehtuuria. Kyseinen arkkitehtuuri on MVC:n kaltainen arkkitehtuuri ja helposti laajennettavissa. Arkkitehtuuri perustuu tietokerrokseen, businesslogiikkakerrokseen ja käyttöliittymäkerrokseen. (Easy Projects .Net. n.d.)

EP perustuu .NET-ohjelmistokehykseen. Sovellus vaatii alustakseen vähintään MS IIS 5.0 -palvelinalustan Microsoft-arkkitehtuurinsa vuoksi. Sovellus saattaa toimia myös Linux-alustalla, jos käytetään mono-moduulia. Sovelluksen valmistaja ei kuitenkaan luvannut täyttä yhteensopivuutta Linux-alustalle. Tietokantapalvelimeksi EP vaatii MS SQL server 2000–2008:n tai ilmaisen MS SQL Express -tietokantapalvelimen. Palvelimelta vaaditaan myös tuki .NET 2.0 ohjelmistokehyksellä, mikä rajaa pois vanhemmat versiot MS IIS web-palvelimesta. Palvelimen tehovaatimuksiksi EP-verkkosivusto ilmoittaa mi-

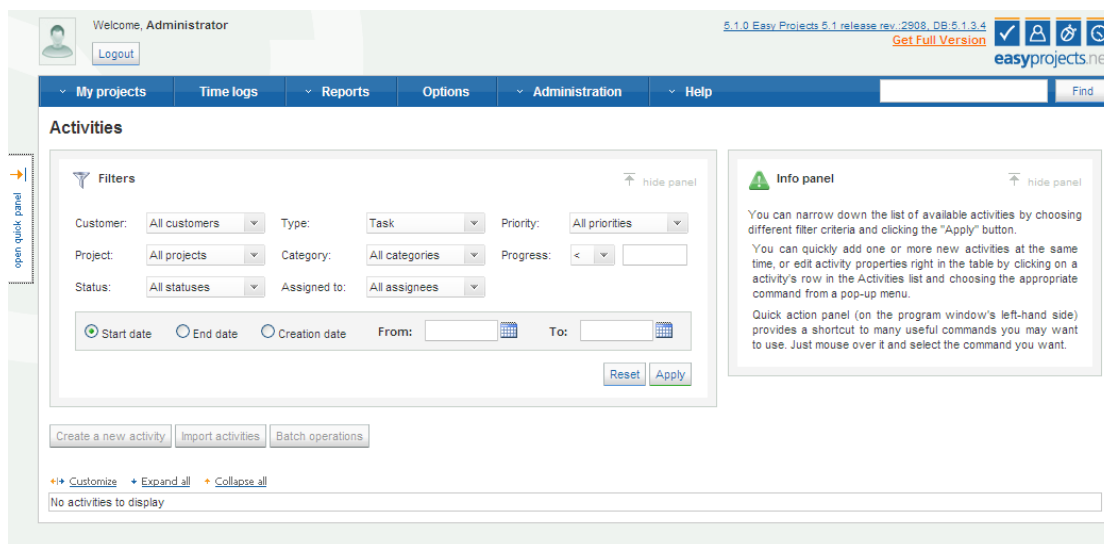
nimissään Intel Pentium-4 2Ghz -prosessorin ja vähintään 1Gt:a Ram-muistia. (Easy Projects .Net. n.d.)

EP:n alustavaatimusten takia hankintakustannukset nousevat hiukan vastaavien sovellusten Linux-alusta hinnasta. Moonsoft.fi-sivuston mukaan Windows 2003 small business server 20 -käyttäjän palvelinkäyttöjärjestelmän lisenssihintana on n. 1200 €. MS Sql Express tietokantapalvelin-ohjelmisto on ilmainen. Lisäksi hankintahintaan vaikuttaa ostettava palvelin-järjestelmä. (Easy Projects . n.d.)

### 6.2.2 Ensivaikutelma

Ensivaikutelma EP:stä on hyvä. Kirjaututtaessa sovellukseen käyttäjälle aukeaa yleiseen tyyliin käyttäjäkohtainen etusivu. Etusivulla on saatavilla käyttäjän tarvitsema informaatio omista tehtävistä ja muista aktiviteeteista. Etusivulla on näkyvillä myös toimintaa nopeuttava hakutoiminto, jonka avulla käyttäjä voi hakea tietokannasta haluamiaan aktiviteetteja.

Usealla sivulla on myös informaatiopaneeli saatavilla sivuston toiminnoista. Käyttäjällä on jatkuvasti saatavilla informaatiota sivun toiminnoista, eikä manuaalia tarvitse selata. Haku- ja informaatiopaneelit ovat myös piilotettavissa sivuilta haluttaessa.



### KUVIO 3. Easy Projects .NET, käyttäjän etusivu

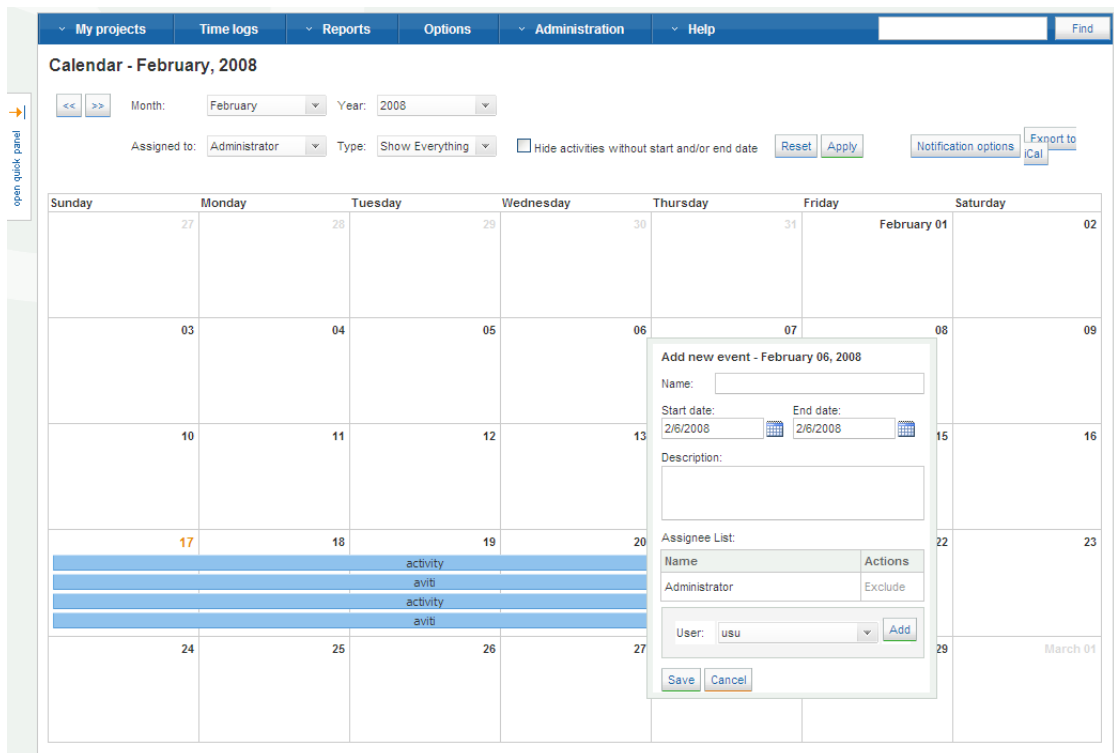
#### 6.2.3 Toiminta

EP:n käyttö alkaa uusien käyttäjien luonnista. Ainoastaan sovelluksen hallintaroolilla oleva käyttäjä voi luoda uusia käyttäjiä. Oletusasetuksien jälkeen sovelluksessa on yksi hallintahenkilö käyttäjänimellä Administrator. Käyttäjälle määritellään näkyvä nimi, kirjautumisnimi, rooli, salasana, sähköpostiosoite ja viikoittainen tuntimäärä. Käyttäjän voi myös asettaa passiiviseksi ja hänet voi määrittää asiakkaan edustajaksi.

Käyttäjäroolitukset ovat osa sovellusten autentikoinnin toteutusta. .NET-sovelluskehys tarjoaa laajat mahdollisuudet roolitusten tekoon palvelinpäästä ja niiden integroinnin jokaiseen sovellukseen. Useissa sovelluksissa on myös omat käyttäjäroolitukset, kuten EP:ssä. Rooleja voi luoda tarvittavan määrän ja jokaisella roolille voi määrittää kohtuullisen laajasta listasta ominaisuuksia, eli mitä rooli voi tehdä ja mitä ei. Käyttäjää voidaan määrittää vain yhteen rooliin, mikä huonontaa roolipohjaisuuden toimivuutta kyseisessä sovelluksessa. Tarvittaessa jokaiselle käyttäjälle voidaan määrittellä vaikka henkilökohtainen rooli, joka ei kuitenkaan ole suositeltavaa. Projektikohtaiset roolitukset ovat myös mahdollisia projektiin resurssimuokkaus-ikkunasta.

EP mahdollistaa puurakente-tyylisen määrittelyn tehtäville. Kukin tehtävä voi olla virstanpylväs, tehtäväryhmän tehtävä tai muodostaa ryhmän jne. Yhteinen ominaisuus nopeuttaa tehtäväryhmien luomista sekä luo selkeän puurakenteen projektin tehtävistä. Jokainen projekti on mahdollista kopioida tehtävineen, mikä helpottaa projektin asiakirjamallien luomista valmiine tehtävineen.

Projektinseuranta on mahdollista myös kalenterinäköymästä ja Gantt-kaaviosta. Kalenterista on mahdollista luoda tapahtumia haluamalleen päivälle. AJAX-teknologiaa on hyödynnetty useassa paikassa. AJAX tuokin sovellukselle huomattavaa käytettävyyttä lisää ja luo sovelluksesta työpöytäsovellustyyppisen.



KUVIO 4. Easy Project .NET, kalenterinäköymä

Raportit on mahdollista luoda joko Excel tai PDF-tiedostomuodossa. Raporttien luontiin voidaan käyttää erillistä raporttien rakennustyökalua, jolla raportti on mahdollista kustomoida haluamukseen. Raporttien luonnissa on mahdol-

lista tuottaa raportteja laskutukseen, asiakkaalle, yleisiä raportteja, Gantt -kaavio ja tuntiseurantaraportteja.

Printer-friendly version Save and run Save Cancel

**Primary fields**

Report name:

Current data source:

Data source:

Font:

Font Size:

Bold:

Italic:

Underline:

**Info panel** ↑ hide panel

Give a name to the report.

To start adding fields to your report, first choose a data source from the respective drop-down list, and click 'Select' to apply your selection. Think of a data source as a pool of interlinked data fields. For example, the All Activities (hierarchical) data source offers a full assortment of activity- and project-related fields and allows producing reports with a hierarchical view of activities.

Only one data source can be used per report.

In the Report fields below, select a field to add from a drop-down list. If you add a field in error, there is an option to exclude it. To edit a field, click Edit. The editing options include Width, Align, and Filter. The first of the three is used to set a width for a field that fits best. The Align option allows you to left-align, center or right-align the value in the field. The Filter option is context-sensitive and offers you an opportunity to set a filter whereby you can choose what items from a selected set to include in your report.

Add as many report fields as may be needed, and click Save and run to generate the report.

**Report Fields**

Name	Width	Align	Filter	
<b>Project ID</b>	Not set	Left	None	<input type="button" value="Exclude"/> <input type="button" value="Edit"/>

KUVIO 5. Easy Project .NET, kustomoitujen raporttien luontilomake

#### 6.2.4 Soveltuvuus

EasyProjects.Net soveltuu projekteihin, joissa kollaboraatio ominaisuudet menivät resursoinnin edelle. Sovellusta voidaan kuvailla skaalautuvaksi erityyppisiin projekteihin. Sovellus soveltuu erinomaisesti IT-projekteihin, mutta se on käyttökelpoinen myös muun tyyppisiin projekteihin.

Sovelluksen helppokäyttöisyys puoltaa sen valintaa yrityksiin tai yhteisöihin, joissa käyttäjien tietotekninen osaamistaso ei ole korkea. Sovelluksen sivuston mukaan se soveltuu käytettäväksi lähes kaikenlaisissa ympäristöissä. Vahva Windows-pohjaisuus kuitenkin rajaa osaksi ulos yritykset, joissa käytettävät ohjelmistot pohjautuvat Unix-järjestelmiin.

### 6.2.5 Yhteenveto

Easy Projects .NET -sovellus on suunniteltu yksikertaiseen projektinhallintaan. Sovelluksessa ei ole ns. turhia ominaisuuksia. Valikot on toteutettu helposti käytettäviksi, käyttöliittymä on selkeä ja sovelluksen toiminta ripeää. Käyttäjien monipuolinen roolipohjainen hallinta on ehdottomasti yksi parhaimpia testiryhmässä.

Sovelluksen henkilöstöresurssien seuranta on sen heikkous. Henkilöstön resursointi on hankalaa ja sovellus ei ilmoita, onko henkilölle yli-allokoitu työtä, vai onko henkilöllä lainkaan tehtäviä. Aktiivisella seurannalla henkilöstöresursointi on kuitenkin mahdollista.

Sovelluksen kustannukset kohdeyritykselle koostuvat seuraavista muuttujista: palvelin Fujitsu Siemens primergy n. 2 200 €. Windows server 2003 (standard edition) palvelinkäyttäjärjestelmä n. 1 200 €. Microsoft SQL Server Express edition ilmainen. Easy Project .Net-lisenssit käyttäjälukumäärän mukaan \$ 149/käyttäjä, suuremmista määristä myönnetään alennusta. Haluttaessa lähdekoodi erillisenä korvauksena \$ 2 500. Easy Projects .Net -sovelluksen kokonaiskustannukset 30 aktiivisille käyttäjälle olisivat noin (\$ 4 470) n. 2 900 € + palvelinalusta käyttöjärjestelmiseen 3 400 €.

### 6.3 TinyErp

TinyErp on GPL-lisenssillä varustettu OS-sovellus eli käytännössä ilmainen. Sovellus on suunniteltu yrityksen hallintajärjestelmäksi, joka tunnetaan nimellä ERP (Enterprise resource planning). Projektinhallinta on rajattu omaan osiinsa tinyErp-sovelluksessa. Sovelluksesta löytyy useita eri alueita ja työkaluja yrityksen sisäisten asioiden hallintaan. Sovelluksessa on seuraavat osiot: CRM, resurssien hallinta, materiaalin hallinta, kirjanpito/laskutus, projektinhallinta, osto- ja myyntitoiminnot.



### 6.3.1 TinyErp, asennus ja käyttöönotto

TinyErp on käyttöjärjestelmäriippumaton sovellus. Sovellus toimii Windows XP -, Windows 2000 -, Unix- ja Linux-ympäristöissä, mukaan lukien Unixiin perustuva Mac OS X. Linux-jakeluversioista yhteensopiviksi oli listattu CentOS, Debian, Gentoo, Mandrake, Mandriva, Suse, FreeBSD ja Ubuntu. Tämä ei kuitenkaan tarkoita sitä, ettei sovellus toimisi muissa jakeluversioissa. tinyErp- tuotteesta löytyy myös työpöytäsovellus, jolla tiedon hallinnointi on mahdollista.

TinyErp vaatii toimiakseen palvelimelta python moduulin toimiakseen. TinyErp:ssä on oma sisäänrakennettu palvelinsovellus, eikä se vaadi pohjalle muuta HTTP-palvelinta. TinyErp-palvelin käyttää PostgreSQL-tietokantaa. PostgreSQL on myös ilmainen OS-sovellus. TinyErp:n palvelin toimii ja tarjoaa tuen sekä HTTP:lle että HTTPS:lle.

Asennettaessa Windows-ympäristöön tinyErp tarjoaa täydellisen asennuspaketin. Paketti sisältää kaikki sovelluksen vaatimat moduulit. TinyErp vaatii NTFS-tiedostojärjestelmän Windows-alustalla, ja se ei ole FAT- tai FAT32-yhteensopiva. Windows-ympäristössä tinyErp vaatii Linuxin tapaan tuen GTK:lle.

TinyErp sisältää tuen lokalisoinnille ja siinä on valmiina useita kieliä. Täydellinen modulaarisuus helpottaa oman kielipaketin lisäämistä. Lokalisointi pitää sisällään myös tarvittavat valuuttatunnukset ja aika- sekä päivämäärä-asetukset. Suomen kieltä ei ole tarjolla valmiina moduulina.

Sovelluskehittäjille tinyErpin modulaarisuus ja muokattavat komponentit tarjoavat rajattomat mahdollisuudet sovelluksen laajentamiseen. TinyErp sovellus on rakennettu MCV-arkkitehtuuria hyväksikäyttäen, mikä selventää sovelluksen rakennetta huomattavasti.

TinyErp käyttää Turbogears-ohjelmistokehystä. Turbogears tarjoaa sovelluk-

sen käyttöön useita kolmannen osapuolen moduuleita.

### 6.3.2 Ensivaikutelma

TinyErp on luonnostaan laaja sovellus. Käyttöliittymä koostuu sivun yläosas-  
sa olevasta navigointivalikosta. Navigointivalikko on vapaasti muokattavissa.  
Navigointilinkkien järjestys, määrä ja nimi ovat muutettavissa vapaasti. Uu-  
den projektin luominen vaatii käyttäjältä vain kaksi painallusta, mikä on eh-  
dottomasti yksi käytettävyyttä parantavista ominaisuuksista.

The screenshot shows the 'Tasks' interface in TinyErp. At the top, there are navigation buttons: Search, Form, Calendar, Graph, and a dropdown arrow. On the right, there are buttons for 'REPORTS' (Gantt Represent), 'ACTIONS' (Bill tasks), and 'Filter' (New). Below the navigation is a search form with 'Basic Search' and 'Advanced Search' tabs. The search form includes fields for 'Task summary', 'Assigned to', and 'Project'. Below the search form are 'Filter' and 'New' buttons. The main area contains a table of tasks with the following columns: Sequence, Task summary, Project, Assigned to, Deadline, Planned hours, Effective hours, Priority, and State. The table shows one task with the following details:

Sequence	Task summary	Project	Assigned to	Deadline	Planned hours	Effective hours	Priority	State
10	abstract	αιμστολογικό	Demo User	2008-02-19 00:58:30	00:00	00:00	Medium	Open

At the bottom of the table, there are input fields for 'Planned hours' (0.00) and 'Effective hours' (0.00). Below the table are 'Import' and 'Export' buttons, and a pagination control showing '<< First < Previous [1 - 1 of 1] Next > Last >>'.

KUVIO 6. TinyErp-projektintehtävälistaus

TinyErp käyttöliittymä on graafiselta näkymältään karu, mutta selkeä. Projek-  
tinhallinta-osion näkymässä ensimmäisenä vastaan tulee lista tehtävistä. Pro-  
jektin tehtävät on listattu taulukkoon, jossa ovat AJAX:lla toteutetut välilehdet  
ja sarake järjestely. Gantt-näkymä saadaan helposti esille valitsemalla tehtä-  
vät, joista halutaan graafinen tulostus. Käyttöliittymä on helposti sisäistettä-  
vissä, ja se toimii kussakin osiossa samalla periaatteella.

### 6.3.3 Toiminta

TinyErp-sovelluksen käytön opettelu vaati käyttäjältä aikaa. Sovelluksen use-  
at lohkot vaikeuttavat selkeän hierarkkisen kuvan saantia. Jokainen toiminto  
on usean painalluksen päässä. Etenkin järjestelmän valvojan näkymä on erit-

täin laaja. Normaalikäyttäjän kirjautuessa näkymä selkeytyy huomattavasti, mihin syynä ovat näkymästä puuttuvat hallintatyökalut.

Jokaiselle projektille määritellään luontivaiheessa projektipäällikkö. Projektipäälliköllä on oikeus muokata projektia täydellisesti. Projektille määritellään jäsenet eli projektiryhmä. Jokaisella projektiin kuuluvalla jäsenellä on oikeus nähdä projektin tiedot ja tehtävät.

TinyErp sisältää kohtuullisen laajat resursointi työkalut, mutta muutama parannus olisi kuitenkin tehtävä. Ajanhallinta työkalu on kohtuullisen selkeä ja laaja. Projektiryhmän jäsenellä on oikeus merkitä tehdyt työtunnit hänelle assosioituihin tehtäviin. Jokaisen työntekijän tehdyt tunnit, tehtävät ja yhteystiedot ovat saatavissa työntekijän tietojen alta.

The screenshot shows the 'Project' management interface in TinyErp. The project name is 'Testproj'. The project manager is 'Administrator' with a priority of 1. The project started on 2008-02-10 with planned hours of 00:00 and working hours set to 'Default timesheet'. The project should end on 2008-02-19, with 00:00 hours spent and a state of 'Open'. There is an 'Active' checkbox and a 'Toggle activity' button. The 'Project's members' table lists 'Administrator' (login: admin) and 'Demo User' (login: demo).

Name	Login
Administrator	admin
Demo User	demo

## KUVIO 7. TinyErp, projektinäkymä

Projektinäkymän käytettävyys on kohtuullinen. Näkymä perustuu välilehdillä toteutettuun projektikohtaiseen hallintapaneeliin. Hallintapaneelissa ovat seuraavat välilehdet:

- Projektinhallinta: projektihallinta-välilehden alla ovat projektin yleiset asetukset ja tiedot.

- Asiakkaan tiedot: välilehdellä on asiakkaan ja laskutuksen informaatio.
- Projektin tehtävät: välilehdeltä löytyy projektin tehtävät taulukko. Taulukkonäkymässä on tehtävän nimi, vastuuhenkilö, deadline, tunnit ja tehdyt tunnit.
- Muistiinpanot: välilehdellä ovat mahdolliset muistiinpanot projektista.

TinyErp sisältää laajat ominaisuudet raporttien rakentamiselle. Raportti ja sen sisältö ovat muokattavissa vapaasti. Raportin tiedostoformaattit ovat CSV ja Excel-tiedostoformaatti.

#### 6.3.4 Soveltuvuus

TinyErp-sovellus soveltuu yritykseen, jolla on tarve saada kokonaisvaltainen hallintajärjestelmä useille osa-alueille. Sovelluksen laajat ominaisuudet hallita yritystoiminnan kokonaisuuksia ovat sen vahvuus. Pelkkään projektinhallintaan sovellus on liian monimutkainen ja laaja.

Sovelluksen käyttöönotto vaatii käyttäjältä panostusta sen toimintojen opetteluun. Sovellusta ei voi helposti suositella yritykseen, jossa tietotekninen osaaminen on heikkoa. TinyErp:n kokonaisuus on liian pieni suuren yrityksen käyttöön ja liian laaja pienien yritysten käyttöön.

Mikäli kohdeyrityksessä on korkea tietotekninen osaamistaso, voi TinyErp-sovellusta suositella käyttöön. Sovellus on vapaasti muokattavissa omiin tarpeisiin sopivaksi. Sovellukseen on saatavilla myös paljon ilmaisia moduuleita sen muuttamiselle.

### 6.3.5 Yhteenveto

TinyErp on laaja sovellus pelkäksi projektinhallintatyökaluksi. Laajuus on juurikin sovelluksen suurin heikkous. TinyErp on erinomainen sovellus pk-yrityksen hallintajärjestelmäksi, mikäli tarvetta on jokaiselle osa-alueelle. Pelkästään projektinhallinta työkaluksi sovellus osoittautuu helposti kankeaksi ja sekavaksi.

Testauksen aikana sovellus antoi lukemattomia kertoja virheilmoituksia, joista voidaan päätellä sovelluksen olevan toistaiseksi hyvin keskeneräinen. Virheilmoitukset eivät olleet selkeitä ja lokiin niistä ei tullut minkäänlaista merkintää. Tämä ominaisuus alentaa sovelluksen käyttöarvoa huomattavasti. PostgreSQL-tietokannan toimintavarmuus oli testin aikana erinomainen.

Käyttöönottokustannukset ovat sovelluksen lisenssin kannalta ilmaiset. Käyttöönottokustannuksiksi muodostuvat käytettävän palvelimen hankintakustannukset. Jos käytetään referenssipalvelinta, kustannukset ovat n. 2 200 €. Palvelinalustaksi suositellaan Linux-palvelinta, jonka kustannukset käyttöönotossa ovat 0 €.

## 6.4 ActiveCollab

ActiveCollab on a51 development -nimisen yrityksen tuote. Tuotteen vanhat julkaisut olivat vielä OSS-lisenssillä varustettuja, mutta versiosta 1.0 asti tuote on ollut kaupallinen. ActiveCollab tarjoaa kahdenlaista kaupallista lisenssiä Corporate ja Small Biz, joiden hinnat ovat 399 ja 199 USD. Lisenssit poikkeavat toisistaan muutamilla ominaisuuksilla ja edullisemmassa paketissa seuraavat ominaisuudet puuttuvat: kalenteri, tehtävät, ajan seuranta, sivut. Kummatkin lisenssityypit ovat pysyviä, eli laskutus tehdään vaan kertaluontoisesti. Lisenssit pitävät sisällään myös yhden vuoden tuen ja päivitykset sovellukseen. (ActiveCollab. n.d.)

Erityisen positiivisen kuvan antaa activeCollab sovelluksen OSS-pohjaisuus. Lisenssien mukana toimitetaan lähdekoodi kokonaisuudessaan. Lähdekoodi on myös vapaasti muokattavissa sovelluksen hankinnan jälkeen. Sovellus toimitetaan asennuspakettina, joten asiakkaan on asennettava se itse omalle palvelimelleen, tai vaihtoehtoisesti asiakas voi vuokrata palvelimen kolmantelta osapuolelta.

#### 6.4.1 ActiveCollab-sovelluksen asennus

ActiveCollab on toteutettu PHP-ohjelmointikielellä. Vähimmäisvaatimukset asiakkaan alustalle ovat Apache web -palvelin joko PHP4- tai PHP5-tuella. Sovellus käyttää tietovarastonaan MySQL-tietokantamoottoria, ja MySQL:n tulee olla vähintään versiota 4.1. Sovellus vaatii myös mysql-, ctype- ja tokenizer-nimiset laajennukset aktivoituna PHP:n asetuksista. Suositeltiin myös seuraavien laajennusten kytkemistä päälle: gd, mbstring ja iconv. ActiveCollab käyttää myös erittäin paljon AJAX:iin pohjautuvia toimintoja, jotka parantavat osaltaan sovelluksen käytettävyyttä ja vähentävät palvelimen kuormaa.

#### 6.4.2 Ensivaikutelma

Ensivaikutelma activeCollab-sovelluksesta on selkeä. Kirjautuessaan sovellukseen käyttäjä näkee henkilökohtaisen kojelaudan(dashboard). Kojelaudassa käyttäjällä on näkymä viimeisistä muutoksista, myöhästyneistä tehtävistä, uusista tai päivitettyistä tiedoista ja omista tähdeksi merkityistä elementeistä. Käyttäjäkohtainen etusivu nopeuttaa käyttäjän omien tehtävien paikannusta ja tuo nähtäville mahdolliset muutokset. ActiveCollabin omalaatuinen käyttöliittymä kilpailijoihin nähden on sekä positiivinen, että negatiivinen puoli.

Alla olevassa kuviossa on nähtävillä activeCollab-sovelluksen käyttäjäkohtainen etusivu(dashboard). Käyttäjälle generoidaan tarvittavat tiedot saataville nopeine navigaatioineen. Sovelluksen käyttöliittymä, yksinkertaisine navigaa-

toineen on ehdottomasti activeCollab:n vahvuus.

The screenshot shows the ActiveCollab dashboard interface. At the top, there is a navigation bar with the ActiveCollab logo, user profile information, and navigation icons for People, Projects, and Calendar. Below the navigation bar, the dashboard is divided into several sections:

- Dashboard Summary:** Three summary cards: "New / Updated" (no new items), "Late / Today" (3 tasks late), and "Starred" (2 objects starred).
- Active Projects:** A section for the "mycomp test project" showing the leader, group, and progress (0 tasks done of 7).
- Recent Activities:** A list of recent activities from yesterday, including tasks, timerecords, and comments.

Yesterday		
Task: taski2	In mycomp test project	Created by [user]
Timerecord: 2.5 hours by qwe	In mycomp test project	Created by [user]
Task: this is task	In mycomp test project	Created by [user]
Checklist: Checklist this is a checklist under milestone	In mycomp test project	Created by [user]
Timerecord: 5 hours by did work	In mycomp test project	Created by [user]
Comment: Comment on summary herajjee	In mycomp test project	Created by [user]

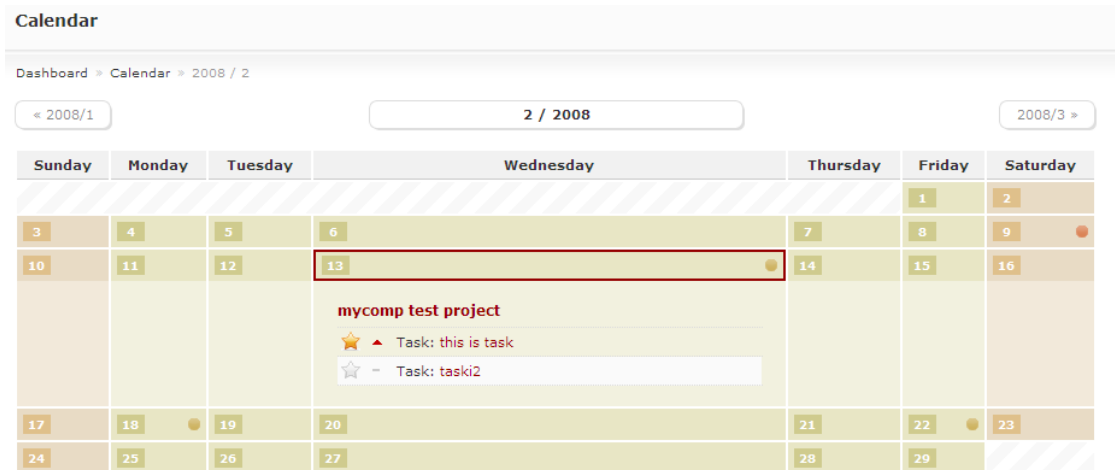
KUVIO 8. ActiveCollab-sovelluksen kojelauta

### 6.4.3 Toiminta

Sovelluksen toimintaperiaate on selkeä. Sovellus toimii täysin projektilähtöisesti. Ensimmäinen askel on luoda käyttäjät ja asettaa käyttäjille roolit projektikohtaisesti. Tämä ominaisuus helpottaa projektien hallintaa. Yksi käyttäjä voi kuulua projektikohtaisesti moneen rooliin. Käyttäjä voi olla projektissa projektipäällikön roolissa, ja toisessa projektissa hänellä on huomattavasti vähemmän oikeuksia. Roolipohjainen käyttäjäautentikointi voi olla ehdoton välttämättömyys projektihallintasovelluksen kollaboratio-ominaisuudet huomioon ottaen.

Projektinluonnin jälkeen projektille määritellään virstanpylväät(milestone). Virstanpylväät toimivat projektia jakavina osioina, joita osassa sovelluksissa käsitellään termillä vaihe (phase). ActiveCollab sovelluksessa virstanpylvään

sisään tulee tehtävälista (checklist), jonka alle määritellään tehtävät (tasks). Rakenne on vahvasti hierarkkinen: projekti->virstanpylväs->tehtävälista->tehtävä. Jokainen virstanpylväs voi sisältää useita tehtävälistoja ja jokainen tehtävälista useita tehtäviä. Rakenteesta ei valitettavasti saa puunäkymää, joka selventäisi paljon sovelluksen hierarkiaa.



KUVIO 9. ActiveCollab-sovelluksen kalenteri

Kuviossa 6 on esitetty activeCollab-sovelluksen kalenterinäkymä. Kalenteri päivittyy automaattisesti käyttäjän tehtävien mukaan. Klikkaamalla kalenterin päivää, jossa käyttäjällä on merkitty päivämäärä tehtävään, aukeaa mahdollisuus merkitä tehtävä tähdellä tai käyttää hyperlinkkiä tehtävään. Kalenteri-ominaisuus helpottaa tehtävien seuranta ja toimii täysin projektikohtaisesti.

#### 6.4.4 Soveltuvuus

ActiveCollab soveltuu parhaiten pieniin nopeisiin projekteihin. Sovellus tukee ainoastaan projektien ja tehtävien hallintaa. ActiveCollab ei sovellu rankasti hierarkkiseen projektinhallintaan, jossa jokainen asia käydään tarkasti läpi. Sovellus on enemmän vapaamuotoisemman työyhteisön työväline ketterine ominaisuuksineen.

ActiveCollab-sovellusta voidaan myös kutsua projektinhallintasovellukseksi,



joka omaa hyvät kollaboraatio-ominaisuudet. Tiedon ja tehtävien jako muiden käyttäjien kesken mahdollistaa yrityksen sisäisen viestinnän sovelluksen kautta. Sovellusta voidaan suositella vapaamman työyhteisön välineeksi hallita tehtävälähtöisesti projekteja.

#### 6.4.5 Yhteenveto

ActiveCollab-sovellus soveltuu osaavan projektiryhmän käyttöön. Se ei luo projektipäällikölle mahdollisuutta seurata resurssien kuormitusta tai resurssien ylityöllistämistä millään tavalla. Projektiryhmänjäsenellä on mahdollisuus merkitä työskenneltyjä tunteja, mutta niitäkään ei voi määrittää tehtäväkohtaisesti. Resurssien seuranta vaatisi sovellukselta paljon lisäominaisuuksia ja nostaisi sovelluksen hintaa huomattavasti.

Sovellusta on helppo suositella pieniin projekteihin osaavalle ryhmälle, jolle resurssien seuranta ei ole tärkein asia. Teollisuuden käyttöön sovellus on hieinan latteaa, mutta toimii erittäin hyvin vapaammassa projektiympäristössä. Monien käytännöllisten ominaisuuksien, edullisten kustannusten ja selkeän käyttöliittymän ansiosta sovellus on hinta-laatu-suhteeltaan erinomainen.

ActiveCollab sovelluksen käyttöönotto aiheuttaa seuraavat kustannukset: palvelin Fujitsu Siemens Primergy n. 2 200 €. Palvelinkäyttöjärjestelmä CentOS distro ilmainen. Sovelluslisenssi yrityksen tarpeesta riippuen Corporate-lisenssi (\$ 399) 259 € tai Small Biz (\$ 199) 129 €. Sovelluksen käyttöönotto-kustannuksiksi tulisi kokonaisuudessaan palvelin 2 200 € ja lisenssi tarpeesta riippuen.

#### 6.5 TRAC

Trac-projektinhallintaohjelmisto on Edgewall-yhteisön kehittämä tuote. Edgewall-yhteisö julkaisee erityyppisiä web-sovelluksia sähköpostiohjelmista Python ohjelmointikielikirjastoihin. Yhteisön tunnetuin tuote on Trac ja sen

ympäriille rakennetut laajennukset.

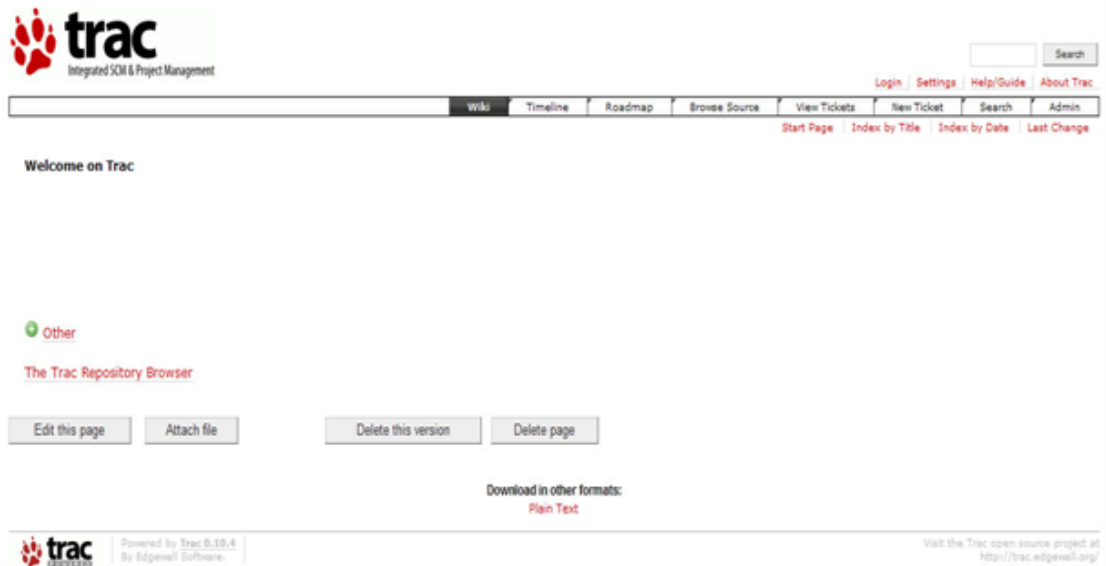
Trac on projektinhallintaohjelmistona kevyt ja kehittäjien ajatuksena on ollut luoda minimalistinen täysin web-ympäristöön implementoitunut sovellus. Tracin toiminta perustuu wiki-tyyppisen alustan hyödyntämiseen, eli siihen on mahdollista lisätä sisältöä paikasta ja ajasta riippumatta. Wikin sisällön tuottamiseen ei tarvita mitään erikoisosaamista, sillä wikin oma Python-alustamoottori hoitaa uuden tekstin muuntamisen oikeaan muotoonsa lisäämällä siihen tarvittavat html-tagit.

### 6.5.1 Tracin asennus

Trac-projektinhallintaohjelmisto tarvitsee toimiakseen paikallisen web-palvelin ohjelmiston sekä raudan. Lisäksi trac tarvitsee palvelimelle asennettavan Python-ohjelmistokielituen. MySQL-tietokantatuki tracissä on vielä beta-testissä, eikä sen käyttöä suositella. MySQL:n tilalla tietokantamoottorina tulisi olla PostgreSQL tai SQLite.

### 6.5.2 Ensivaikutelma

Wiki-pohjaisuus on selkeästi esillä. Valikot löytyvät helposti ja ne ovat hyvin yksinkertaistettuja. Trac ei kuitenkaan toimi ihan "out of the box" menetelmällä, ja wiki-perusteisen ympäristön saattaminen käyttökuntoon vaatii paljon hiomista. Ensimmäisenä kannattaa tutustua käyttäjäoikeus asetuksiin ja tehdä alustavia testejä, jotta tulee varmistettua oikeuksien asetusten oikeellisuus. Esimerkkinä voidaan mainita se että, ilman oikeuksien määrittämistä voi kuka tahansa sivuille päässyt henkilö tyhjentää etusivun kaikesta tiedosta, jota siihen on lisätty.



KUVIO 10. Trac-sovelluksen etusivu

### 6.5.3 Toiminta

Tracin toiminta perustuu projektinhallintaan virstanpylväiden (milestone) ja tehtävien (ticket) avulla. Projektiin määritellään tietyt virstanpylväät, joiden sisälle määritellään alustavat tehtävät. Tehtäviä voidaan lisätä milloin tahansa projektin etenemisen aikana. Tehtävät tracissä ovat avainasemassa ja ne voivat toimia erityyppisinä elementteinä. Elementtejä ovat tavalliset projektintehdävä, ominaisuuspyyntö, vika-raportti ja ohjelmistotukipyyntö.





Tehtävän määrittäminen. Tämän välilehden avulla luodaan uusi tehtävä, joka sitten sijoitetaan oikean virstanpylvään alaisuuteen. Virstanpylväät toimivat tehtävälistoina (checklist).

Your email or username:

Short summary:

Type:

Full description (you may use [WikiFormatting](#) here):

**B I A**    

**Ticket Properties**

Priority: <input type="text" value="normal"/>	Milestone: <input type="text"/>
Component: <input type="text" value="general"/>	Version: <input type="text"/>
Severity: <input type="text" value="normal"/>	Keywords: <input type="text"/>
Assign to: <input type="text"/>	Cc: <input type="text"/>





I have files to attach to this ticket

## KUVIO 11. Trac-sovelluksen tehtävän määrittäminen

Timeline-välilehti kuvaa päivittäiset tehtävät ja niiden tilan. Välilehdellä näytetään myös edellisen päivän tehtävät.

## Timeline

**02/12/2008:**

-  18:46 **Ticket #6815 (defect) closed**  
wontfix: You're using the [tinymcewiki plugin](#). This is ...
-  18:40 **Ticket #6817 (enhancement)**  
[patch] Showing related changesets on a ticket
-  16:56 **TracMercurial edited by anonymous**  
mercurial version requirement ([diff](#))
-  13:40 **Ticket #6785 (defect)**  
fixed

## KUVIO 12. Trac-sovelluksen aikajana

Roadmap-välilehdellä kuvataan virstanpylväiden saavuttamista selkeällä prosenttiluvulla sekä sitä kuvaavalla palkilla.

## Roadmap

**Milestone: 0.10.5**

*No date set*



Bug fixes and minor improvements.

### KUVIO 13. Trac-sovelluksen roadmap

Raportointi-ominaisuudet tracissä ovat todella kattavat. Valmiina raportteina löytyy useita erilaisia vaihtoehtoja projektin tilan tarkasteluun.

### Available Reports

This is a list of reports available.

Report	Title
{1}	All Active Tickets
{2}	Active Tickets for Ticket Module
{3}	Active Tickets for Wiki Module
{4}	Active Tickets for Browser Module
{5}	Active Tickets By Version
{6}	Active Tickets By Milestone
{9}	Assigned, Active Tickets by Owner
{10}	Assigned, Active Tickets by Owner (Full Description)
{11}	All Tickets By Milestone (Including closed)
{12}	My Tickets
{13}	Active Tickets, Mine first
{14}	List tickets where priority is \$PRIORITY (Dynamic variable test)
{20}	Active tickets with no Milestone, by last modified date
{21}	Tickets awaiting user feedback, by last modified date
{22}	Active tickets for 0.10-stable, by last modified date
{23}	Active tickets for 0.11, by last modified date
{24}	Active tickets for future releases, by last modified date
{25}	Active tickets for 0.11-stable, by last modified date

### KUVIO 14. Trac-sovelluksen raportit

Trac-sovellus antoi pienen suojan tiedolle wiki-pohjaisuudestaan huolimatta. Tiedonmuokkauslomakkeen ollessa auki ei ole mahdollista tallentaa tietoa, jos toinen käyttäjä on muuttanut tietoa muokkauksen aikana. Ominaisuus estää tiedon päällekkäismuokkauksen, mitä voidaan pitää hyvänä asiana. Käyttäjällä on mahdollisuus tallentaa muokkaamansa tieto leikepöydälle ja avata muokkaustila uudelleen.

#### 6.5.4 Soveltuvuus

Tracin käyttäjäkunta koostuu monista erikokoisista yrityksistä. Mainitsemisen arvoisia ovat esimerkiksi NASA:n FUN3D-laboratorio. Kuitenkaan mitään suuria kokonaisuuksia Tracillä ei hallita. Trac soveltuu parhaiten pienten projektiryhmien työkaluksi, joilla on osaamista myös Tracin käyttökuntoon laittamisen vaatimasta web-ohjelmoinnista. Tracin suosion syyksi muihin pieniin projektinhallintasovelluksiin verrattuna voidaan todeta sen wiki-pohjaisuus. Tracin oma wiki toimii monessa yrityksessä yhteydenpitovälineenä sidosryhmien välillä.

#### 6.5.5 Yhteenveto

Trac on hyvä valinta pienen projektien hallintaan. Wiki-pohjaisuus ei ole kaikkien mieleen, mutta Tracissä se on toteutettu hyvän maun mukaisesti. Käyttäjäoikeuksien rajaamisen toteuttaminen on tapahtunut erittäin heikosti johtuen wiki-tyyppisestä lähestymistavasta ohjelmiston toteutukseen. Python ohjelmointikieli ei ole kaikkein yleisin valinta toteuttaa web-pohjaisia ohjelmistoja. Ylimääräistä työtä se aiheuttaa kuitenkin vain käyttöönoton aikana.

Hankintakustannukset Tracissä muotoutuvat pelkästään palvelinalustasta sekä valittaessa Windows Server sen lisenssistä. Käytettäessä ilmaista Linux-käyttöjärjestelmää kustannukset olisivat 2 200 € ja Windows-lisenssin kanssa n. 3 400 €.

### 6.6 SugarCRM

SugarCRM ei ole pelkästään projektinhallintaohjelmisto, vaan kokonaisvaltainen yrityksen asiakkuudenhallintajärjestelmä CRM (Customer Relationship Management). Tutkimuksessa tarkastellaan vain sovelluksen projektinhallinta ominaisuuksia.

SugarCRM-yritys on perustettu vuonna 2004 ja nykyään yrityksen palveluksessa on yli 150 henkilöä. SugarCRM-tuoteperhe sisältää seuraavat tuotteet: SugarCRM Open Source, Sugar professional, Sugar Enterprise, Sugar Fast-track, Sugar Network sekä Sugar University. Sugar professional- ja Enterprise-versiot toimivat asiakkaan omalla palvelimella. Enterprise-versio on tarkoitettu suurien yritysten käyttöön ja se sisältää tuen Oracle-tietokantamootorille, laajemman sql-raportoinnin ja offline-päätteen. SugarCRM OpenSource sisältää vain CRM-ominaisuudet. (SugarCRM:n verkkosivut. n.d.)

Suomessa SugarCRM sovelluksen käyttöönotosta vastaa Antamis Finland Oy. Antamis on Brysselissä toimivan Antamis Sprl -ohjelmistoyrityksen tytäryhtiö. (Antamis Finland Oy:n verkkosivut. n.d.)

#### 6.6.1 Asennus

SugarCRM ohjelmistona vaatii toimiakseen Apache Web palvelinsovelluksen, PHP tuki ja MySQL tietokannan. Halutessaan asiakas saa yhden asennuspaketin, joka sisältää kaiken edellä mainitut tuotteet. Palvelinlaitteisto tulisi olla nykyaikaisella tasolla, jotta ohjelmisto toimii siedettävällä nopeudella. Asennus on helppoa, kuka tahansa ohjeita lukenut henkilö pystyy asentamaan ja ottamaan käyttöön ohjelmiston vaivatta. Asennusohjeistus löytyy myös suomen kielellä. Sovellukseen saa ladattua kielipaketteja, joista löytyy myös suomi. (SugarCRM:n verkkosivut.)

#### 6.6.2 Ensivaikutelma

SugarCRM, kuten muutkin suuremmat työssä tutkitut sovellukset, on käyttöösi liittymältään erittäin hyvin toimiva ja käytettävyyteen on panostettu. Projektin seuranta helpottavat visuaaliset elementit, kuten Gantt-kaavio. Visuaalisten elementtien avulla projektien tilaa on helppo seurata, kun kyseessä ovat

laajat projektit. Negatiivista taas on se, etteivät Gantt-kaaviossa näy tehtävien nimet vaan pelkästään tehtävien ID-numerot. Tästä johtuen kaavion selkoku-  
kuisuus kärsii.

### 6.6.3 Toiminta

Projektit luodaan aina tietyn kaavan perusteella. Projektille määritellään aloi-  
tus ja lopetuspäivämäärät, projektin prioriteetti sekä projektiin osallis-  
tuvat henkilöt. Myös projektin tyyppi eli status voidaan määrittää projektin suun-  
nitteluvaiheessa. Määrittelyvaiheen jälkeen asetetaan tehtävät ja virstanpyl-  
vää sekä riippuvuudet toisiin tehtäviin. Sovelluksen avulla on myös mahdol-  
lista luoda kokouskutsuja ja lähettää sähköpostia sidosryhmien välillä.

Projekti-välilehdelle siirryttäessä alkunäkymässä avautuvat hakuominaisuu-  
det, joiden avustuksella etsitään olemassa olevia projekteja. Välilehdellä voi-  
daan myös suorittaa uuden projektin aloitus.

#### Projects: Home

Basic Search
Advanced Search

Only my items

Search
Clear
| Saved Searches
--None--

#### Project List

Name	Start Date	End Date	Status
<a href="#">Create new project plan for audit</a>	01/01/2000	01/01/2000	Draft

#### KUVIO 15. SugarCRM projektivälilehti alkunäkymä

Uuden projektinluominen on selkeä käyttöliittymä ja käsittää erittäin mitta-  
van valikoiman erilaisia ominaisuuksia, joita projektille voidaan määrittää.



**Project: Testi**

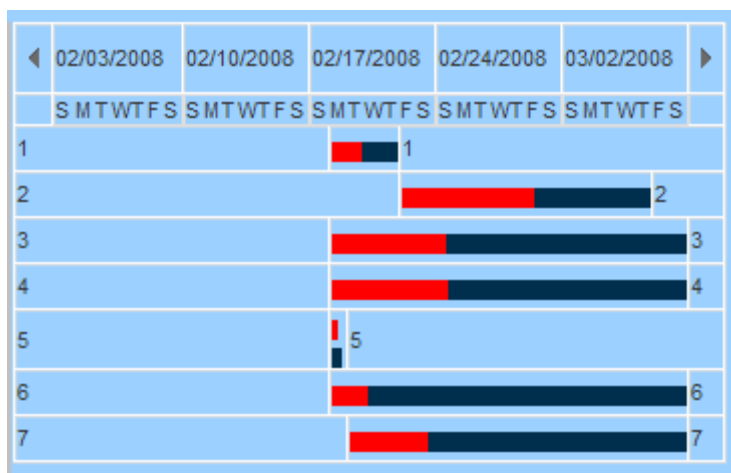
Name:	Testi	Status:	Draft
Start Date:	02/03/2008	End Date:	02/18/2008
Assigned To:	admin	Team:	(admin)
Priority:	High		
Description:			

- ⌵ Resources
- ⌵ Project Tasks
- ⌵ Activities
- ⌵ History
- ⌵ Contacts
- ⌵ Accounts
- ⌵ Opportunities
- ⌵ Quotes
- ⌵ Cases
- ⌵ Bugs
- ⌵ Products
- ⌵ Project Holidays

## KUVIO 16. SugarCRM uuden projektin luominen

Projektin kokonaistilan hahmottaminen Gantt-kaavion avulla on helppoa.

Myös uusien tehtävien luominen voidaan toteuttaa tällä välilehdellä.



## KUVIO 17. SugarCRM, Gantt -kaavio

#### 6.6.4 Soveltuvuus

SugarCRM:ää ei voida suositella pelkästään projektinhallintaohjelmistoksi. Projektinhallintaominaisuudet toimivat hyvin yhteistyössä muiden sovellukseen liittyvien moduulien kanssa. Suomi-kielipaketti ja suomenkielinen tuki ovat hyvä lisä, joka puuttuu monista projektinhallintasovelluksista. Jos yritys tarvitsee asiakkuudenhallintasovellusta ja projektinhallintasovellusta samassa paketissa, on SugarCRM varteenotettava vaihtoehto.

#### 6.6.5 Yhteenveto

SugarCRM on erittäin monipuolinen ohjelmisto, jonka avulla pystytään hallitsemaan muutakin kuin projekteja. Ohjelmisto sisältää kokonaisvaltaisen pake-  
tin pk-yrityksen asiakkuudenhallintaan. Ominaisuuksien laajuus tuo tulles-  
saan myös negatiivisia puolia. Ohjelmiston lukuisat ominaisuudet hidastavat huomattavasti sen toimintaa.

SugarCRM:n OpenSource hankintakustannukset ovat kertaluontoisia, eli jo aikaisemmin mainittu palvelin 2 200 €. Windows Server -alustalla kustannukset nousevat kokonaisuudessaan noin 3400 euroon. Halutessa vaihtoehtona ovat myös käyttökustannuksia nostattavat lisenssit: Enterprise 449 \$ tai halvempi Corporate-versio 240 \$.

### 6.7 dotProject

DotProject-sovelluksen alkuperäinen nimi oli dotmarketing.org vuonna 2000, minkä jälkeen sovellus nimettiin uudelleen. DotProject on ilmainen avoimen lähdekoodin sovellus. DotProject on suunniteltu erityisesti projektien kontrollointiin. Sovellusta hallinnoi ja kehittää aktiivinen ryhmä vapaaehtoisia sovel-  
luskehittäjiä. Sovellukseen syntyykin uusia ominaisuuksia ja päivityksiä nopealla tahdilla.

DotProject-sovellus sisältää työkalut projektinhallintaan, aikataulutukseen, seurantaan ja tiimi-kollaboraatioon. Jokainen ominaisuus on kuitenkin pyritty toteuttamaan projektinhallinnan näkökulmasta.

#### 6.7.1 Asennus

DotProject-sovelluksen sivustolla suositellaan alustaksi vähintään Apache 1.3.27 -palvelinsovelluksesta. Sovellus on toteutettu PHP-kielellä ja sovellus vaatii vähintään PHP-version 4.2. Tietokantamoottorina sovellus käyttää MySQL-tietokantaa. Sovellus voidaan asentaa myös muiden palvelinsovellusten päälle, kuten Mac OS X tai Microsoft IIS. Jokainen alusta vaatii tuen PHP:lle ja MySQL-tietokantamoottorin asennettuna.

DotProjectin asennus on kohtuullisen yksinkertainen. Sovelluksesta on olemassa erittäin kattava dokumentaatio, jossa asennus toimenpiteet käydään läpi askel askeleelta. Sovellus sisältää oman asennus moduulin, joka löytyy sovelluksen kansioista. Asennuksen voi suorittaa tarvittaessa myös SSL:n päällä. SSL:n käyttö on suositeltavaa etäasennusta tehtäessä. Etäpalvelimella on oltava tuki SSL:n käyttämiselle, mikäli halutaan luoda suojattu yhteys.

DotProject-sovellus tukee myös LDAP-verkkoprotokollaa. LDAP:n avulla on mahdollista esimerkiksi integroida käyttäjät AD:sta. LDAP-autentikaatiota käytettäessä säästytään erillisiltä käyttäjälisäyksiltä tietokantaan. Tämä mahdollistaa yrityksen käyttäjätietojen säilymisen yhdessä keskitetyssä pisteessä.

Asennus vaatii käyttäjältä ehkä hieman keskimääräistä laajemmat tiedot eri moduulien ja protokollien toiminnasta. Hyvän dokumentaation asennuksen voi suorittaa hieman kokemattomampikin käyttäjä.

#### 6.7.2 Ensivaikutelma

DotProject vaikuttaa tavalliselta projektihallintasovellukselta. Sovelluksessa

on yksikertainen navigaatio, joka on osaltaan liian laaja. Etusivun avauduttua projektin luonti vaatii kaksi painallusta. Navigointivalikossa sijaitsee osaltaan turhiakin linkkejä, kuten tiedostonhallinta, tehtävät, keskustelut, tehtäväryhmät ja bugit. Jokainen luetteloiduista osioista liittyy projekteihin ja niille ominainen sijoituspaikka olisikin hierarkkisesti projektin alla.

Sovelluksesta saa vaikutelman sen selvästä linjasta projektinhallintasovellukseksi. Käyttöliittymä on arkinen ilman koristeita, tai Web 2.0 tyyllisiä ratkaisuja. Ajaxin puuttuminen olennaisista kontrolleista syö osaltaan sovelluksen käytettävyyttä.

### 6.7.3 Toiminta

DotProject-sovelluksessa on projektinhallintasovellukselle ominaiset toiminnot, kuten projektin luonti, tehtävien määrittäminen ja seuranta.

Projektia luotaessa projektille on määriteltävä pakolliset attribuutit, kuten nimi ja päivämäärät. Projektilla on aina omistaja, joka on myös määritelty projektin päälliköksi. Sovellus mahdollistaa myös budjetin määrittämisen projektille yksinkertaisena lukuna. Sovellus laskee projektin edistymisen aikana todelliset kulut, jotka määräytyvät tehtyjen tuntien ja merkittyjen materiaalikulujen mukaan. Projektin luontivaiheessa projektille määritellään myös tila. Tilaa käytetään osaksi helpottamaan useiden projektien luokittelussa.

Projektia luotaessa sille voidaan määritellä myös seuraavat attribuutit: projektin tyyppi, asiakas, tekijä yritys ja prioriteetti. Listatessa useita projekteja kyseiset attribuutit helpottavat projektien järjestämistä ja löytymistä. Huomattavaa on, että DotProject-sovelluksessa on mahdollista määrittää projektin asiakkaaksi esimerkiksi oma yritys ja suorittajaksi toinen yritys. Yrityksille jotka käyttävät kolmansia osapuolia osana projekteja, kyseinen ominaisuus on tärkeä.

Companies | Projects | Tasks | Calendar | Files | Contacts | Forums | Tickets | User Admin | System Admin

Welcome Admin Person Help | My Info | **Todo** | Today | Logout

**View Project** Search:

projects list : edit this project : organize tasks : reports

**tama on testi**

Details	Summary
Company: Test 1	Status: In Planning
Short Name: tama on te	Priority: normal
Start Date: 28/02/2008	Type: Operative
Target End Date: 15/02/2008	Progress: 7.9%
Actual End Date: <b>17/03/2008</b>	Worked Hours:
Target Budget: \$150000.00	Scheduled Hours: 112
Project Owner: User, Super	Project Hours: 208.8
URL:	
Staging URL:	
Standardized Definition of ROI:	
<b>Description</b>	

Tasks | Tasks (Inactive) | Forums | Gantt Chart | Task Logs | Events | Files

Show:  Incomplete Tasks Only

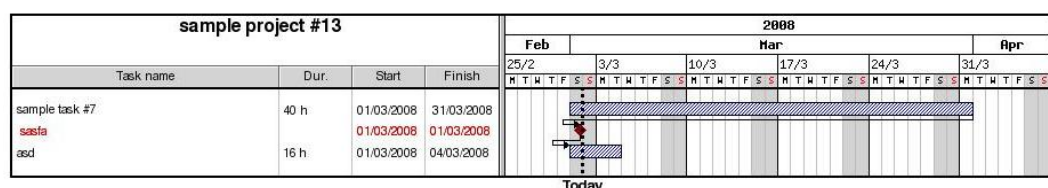
Pin	New Log	Work	P	Task Name	Task Creator	Assigned Users	Start Date	Duration	Finish Date
	Log	55%	↑	taski	admin	admin (35%) (+2)	28/02/2008 11:00	2 days	03/03/2008 10:00
		0%		subtaski	admin	super (30%)	-	1 hours	28/02/2008 13:00
	Log	0%		dependency task	admin	admin (100%) (+1)	28/02/2008 11:45	12 days	17/03/2008 10:45

Key: ■ =Future Task ■ =Started and on time ■ =Should have started ■ =Overdue ■ =Done

## KUVIO 18. DotProject, projektinäkymä

DotProject:n tehtävämäärittely on laaja. Tehtäville on mahdollista määrittellä useita eri attribuutteja ja sidonnaisuuksia. Jokainen tehtävä voi olla virstanpylväs, päätehtävä tai vaihtoehtoisesti lapsitehtävä. Tehtäville on mahdollista määrittellä myös riippuvaisuus toisista tehtävistä, ja näiden määritysten kautta muodostuu myös projektin kriittinen polku. Dynaamiseksi määritelty tehtävä ei voi olla riippuvainen toisesta tehtävästä.

Kalenterista kukin käyttäjä voi nähdä omat tehtävät. Tehtävistä on saatavilla tiedot aloitus- ja lopetuspäivämäärästä. Käytettävyyden kannalta tehtävistä olisi hyvä näkyä aikalinja alusta loppuun kalenterissa, eikä vain tehtävän alkamispäivänä ja loppumispäivänä. Näin tehtävien seuraaminen kalenterin kautta olisi käyttäjälle selkeämpää.



## KUVIO 19. DotProject, Gantt -kaavio

Gantt-kaavio on tulostettavissa projektin tiedoista. Gantt-kaaviota tulostettaessa määritellään haluttu aloituspäivämäärä ja lopetuspäivämäärä, miltä ajalta graafinen kaavio tulostetaan. Gantt-kaavio tulostuu erillisenä kuvana. Kuva on helppo tarvittaessa tulostaa tai tallentaa projektiryhmän käyttöön.

#### 6.7.4 Soveltuvuus

DotProject soveltuu projektinhallintatyökaluksi erinomaisesti. Sovelluksen jokainen komponentti on suunniteltu projektinhallinnan näkökulmasta, eikä sovelluksessa ole projektinhallinnasta irtonaisia osioita.

Sovelluksen monipuolinen tehtävämäärittely mahdollistaa sen käytön myös hierarkisesti rakentuviin projekteihin. Moni sovellus kärsii tehtävien riippuvuuden määrittämisen puutteesta.

Sovellusta voidaan suositella erikokoisille yrityksille projektinhallintasovellukseksi. Sovelluksen arkkitehtuuri vaatii palvelimelta hyvää suorituskykyä, eli mikäli käyttäjiä on paljon tarvitaan myös suhteessa tehokas palvelinalusta.

#### 6.7.5 Yhteenveto

Sovelluksen koristelemattomasta ulkoasusta huolimatta sovellus on kelvollinen työkalu projektinhallintaan. Sovellukseksessa on kuitenkin rajalliset työkalut käyttäjäroolien hallintaan ja asettamiseen. Sovelluksen täydellinen keskittyminen projektinhallintasovellukseksi on nähtävillä sen toiminnoista ja tavasta käsitellä tietoa.

DotProject-sovelluksen hankintakustannukset muodostavat ainoastaan palvelimen kustannukset. Käytettäessä referenssipalvelinta hinta on noin 2 200 €. Palvelinkäyttäjärjestelmäksi suositellaan sovelluksen arkkitehtuurista johtuen Linux-käyttäjärjestelmää.

## 7 TULOSTEN YLEINEN TARKASTELU JA ARVIOINTI

Luvussa käsitellään tutkimuksen tuloksia ja mahdollista jatkokehittämistä.

### 7.1 Tutkimuksen tulokset

Testauksen tulosten perusteella web-pohjaiset projektihallintasovellukset omaavat paljon samankaltaisia ominaisuuksia ja toimintoja. Sovellusten eroavaisuudet toisistaan tulivat parhaiten esille käytettävyyden eroavaisuuksilla. Sovellusten arkkitehtuurien eroista johtuvat poikkeavuudet käytettävyydessä olivat huomaamattomia testausvaiheessa.

Projektinhallintasovellusten määrän ja toimintaperiaatteen eroista johtuvista syistä on hankalaa löytää juuri oikea sovellus kullekin projektityypille. Testissä mukana olleita sovelluksia ei oltu räätälöity tietyntyyppisille projekteille, joten kunkin yrityksen on löydettävä parhaiten soveltuva projektihallintasovellus itselleen. Web-pohjaisten sovellusarkkitehtuurien päällä toimiva sovellus asettaa omat rajoitteensa sovelluksen kokonaisvaltaiselle toiminnalle. Osaltaan web-pohjaisuus poistaa työpöytäsovelluksen rajoitteita.

Projektinhallintasovellusten monimutkainen arkkitehtuuri asettaa monimuotoisia vaatimuksia käyttöympäristölle. Sovellukset vaativat usein monia erilaisia moduuleita toimiakseen täydellisesti. Uusien tekniikoiden käyttö luo sovelluksen arkkitehtuurista monimuotoisen ja vaikeasti muunneltavan kokonaisuuden.

Projektinhallinnan teoriapohjalta tutkittiin mahdolliset ominaisuudet, joita on hyvä seurata projektin kokonaistilanteen hahmottamisen kannalta. Pääasiallisesti kaikki sovellukset täyttivät nämä ennalta määritetyt ominaisuudet, mutta ne olivat toteutettu monesti kehittäjien omasta näkökulmasta.

## 7.2 Jatkotutkimusmahdollisuuksia

Projektinhallintasovellusten alueelta voitaisiin tutkia projektinhallintasovelluksia, jotka on räätälöity tietyntyyppisiin projekteihin. Tutkimusta voitaisiin laajentaa myös ASP-periaattella toimiviin projektinhallintasovelluksiin, sillä nämä sovellukset soveltuvat paremmin yrityksille, joilla ei ole omaa sisäistä verkkoa.

Tutkimusta voitaisiin käyttää myös pohjana erilaisien tehtävähallintasovellusten testaukseen. Kyseiset sovellukset sopivat paremmin pienten kampanjatyypisten projektien hallintaan. Näitä ovat projektit, joiden kesto on lyhyt ja yleinen menettelytapa ketterä.

Käyttöliittymätutkimusta voitaisiin syventää kyselyllä ja kartoittamalla, mikä tekee sovelluksesta käytettävän. Käyttöliittymätutkimuksessa, joka on spesifioitu projektinhallintasovelluksiin, saataisiin tuloksia siitä millainen käyttöliittymä soveltuu parhaiten osaamattoman käyttäjän käytettäväksi. Tutkimukseen voitaisiin liittää myös selvitys siitä millainen raportointi järjestelmä palvelisi parhaiten yritysten vaatimuksia.

## 7.3 Pohdintaa

Tutkimuksen tuloksena nähtiin epäsuorasti myös se, ettei ole mitenkään helppoa valita jotakin tiettyä sovellusta toimimaan tietyntyyppisen projektin apuvälineenä. Monesti ensimmäinen ajateltu sovellus ei toiminutkaan halutulla tavalla projektin yhteydessä.

Tärkein saavutus tässä tutkimuksessa oli web-pohjaisten projektinhallintasovellusten ominaisuuksien ja toiminnallisuuksien määrittely. Testatut sovellukset pitivät sisällään toisistaan poikkeavia toiminnallisuuksi. Suurimmat erot sovelluksissa olivat niiden tavat käsitellä projekteihin liittyviä ominaisuuksia.



Tutkimuksen aikana tuli myös ilmi paljon riskejä, joita web-pohjaiset projektin-hallintasovellukset tuottavat toimintaympäristönsä kautta. Web-ympäristössä toimivat sovellukset ovat hyvin alttiita arkaluontoisen materiaalin joutumisessa väärin käsiin. Siksi tietoturvaan onkin kiinnitettävä erityistä huomiota.

Käytettävyyden merkitys kasvoi ennalta arvaamattomaan rooliinsa sovelluksia vertailtaessa. Sovelluksia oli erittäin paljon ja laatu oli toteutukseltaan hyvää monissa sovelluksissa. Tästä syystä sovellusten käytettävyys vaikutti suuressa määrin siihen, mitä sovelluksia lopulta tutkimukseen valittiin. Teorian avulla saatiin myös kasvatettua omaa osaamista projektinhallinnan perusteista, niihin liittyvistä ominaisuuksista sekä kokonaiskuvan hahmottamisen tärkeydestä.

Työn aikana selvisi paljon käsitteiden eroja. Esimerkiksi projektinhallinnan ja tehtävähallinta eroavaisuudet. Sovelluksien tutkimisen aikana saatiin paljon uutta tietoa siitä, miten erityyppisille projekteille valitaan sitä tukeva projektinhallintasovellus. Kokonaisuudessaan työ oli erittäin haastava, mutta samalla erittäin antoisa ja sivistävä.

Työ toteutui suunnitellussa aikataulussa. Alkuvaiheessa teoriaan tutustuminen venytti aikataulua hieman suunniteltua enemmän. Sovellusten testauksessa päästiin takaisin suunniteltuun aikatauluun.

## LÄHTEET

Active collab verkkosivusto. Viitattu 10.2.2008.

[Http://www.activecollab.com/](http://www.activecollab.com/).

Arkansas department of information systems. 2006. Methodology document.

Viitattu 3.2.2008. [Http://pmstats.dis.arkansas.gov/Meth/Meth\\_index.htm](http://pmstats.dis.arkansas.gov/Meth/Meth_index.htm).

Brinck, T., Gergle, D. & Wood, S. 2002. Usability for the Web: Designin web sites that work. Morgan Kaufmann publishers.

Cornford, T. & Smithson, S. 2006. Project Research in Information Systems.

Second edition. Palgrave Macmillan, New York, USA.

Easy Projects .NET verkkosivusto. Viitattu 17.2.2008.

[Http://www.easyprojects.net/](http://www.easyprojects.net/).

Netcraft Web Server Survey. 2007. Viitattu 15.2.2008.

<http://survey.netcraft.com/Reports/200712/index.html>.

Asp.Net Ajax. Viitattu 12.2.2008. [Http://www.asp.net/ajax/](http://www.asp.net/ajax/).

Jablonski, S., Petrov, I. Meiler & C., Mayer, U. 2004. Guide to Web Application and Platform Architectures. Springer 2004.

Keinonen, T. Tuotteen käytettävyys. Viitattu 14.2.2008.

[Http://www2.uiah.fi/projects/metodi/printabl/068.htm](http://www2.uiah.fi/projects/metodi/printabl/068.htm) -> Vuorovaikutteisen tuotteen käytettävyys .

Project perfect. Project Management. Viitattu 2.2.2008.

[Http://www.projectperfect.com.au/wp\\_index.php](http://www.projectperfect.com.au/wp_index.php).

Rissanen, T. 2002. Projektilla tulokseen – projektin suunnittelu, toteutus, motivointi ja seuranta. Gummerus: Jyväskylä.

Tampereen yliopisto verkkosivusto -> Projektisuunnitelma. Viitattu 27.1.2008.

[Http://www.uta.fi/tvema/projektit/projektisuunnitelma.html](http://www.uta.fi/tvema/projektit/projektisuunnitelma.html).

TinyErp verkkosivusto. Viitattu 15.2.2008. [Http://tinyerp.org](http://tinyerp.org).

Web Application Security Consortium (WASC). 2004. Threat classification.

Viitattu 22.2.2008. [Http://www.webappsec.org/projects/threat/](http://www.webappsec.org/projects/threat/) -> Complete document (PDF).

Virtanen, P. 2000. Projektityö. Porvoo 2000: WS Bookwell Oy.

## LIITTEET

LIITE 1. Sovellusten ominaisuusvertailutaulukko.

	Easy Pro- jects.NET	activeCollab	tinyErp	SugarCrm	trac	DotProject
Hinnoittelu	\$79.79/user	\$199/\$398	-(OSS)	-(OSS)	-(OSS)	-(OSS)
Yritys	Logic software	a51	Tiny sprl	SugarCrm Inc.	Edgewall Software	Slackhat
<b>Projektinhallinta</b>						
Tehtävien hallinta						
Resurssit	√	√	√	√	-	√
Tehtävän editymisen seuranta	√	√	√	√	√	√
Tehtävän deadline	√	√	√	√	√	√
Tehtävä riippuvaisuudet	√	√/-	√/-	√	-	√
Aikataulutus						
Kalenteri	√	√	√	√	-	√
Tapahtumat	-	-	-	√	√	√
Gantt-kaaviot	√	-	√	√	-	√
Raportointi						
Statistiikat	√	-	-	√	-	-
Työ kuormitus	√	-	√	√	-	√/-
Kustomoitava	√	-	√	√	√	-
Dokumenttien hallinta	√	√	-	√	√	√
Budjetointi	√/-	-	√	√	-	√/-
Kriittinen polku	-	-	-	√	-	√
Virstanpylväät	√	√	-	√	√	√
Riski-analyysi	-	-	-	-	-	-
<b>Resurssien hallinta</b>						
Resurssien tiedot	√	√	√	√	-	√
Resurssien tuonti	√	-	-	√	-	-
Resurssin seuranta	√	√	√	√	-	√
kulut	√	-	√	√	-	-
Roolitukset	√	√	-	√	-	√
<b>Kollaboraatio</b>						
Tiimikalenteri	-	-	-	√	-	-
Ongelmien hallinta	√	√	-	√	√	√
Raportit	√	-	√	√	√	√/-
Keskustelut	√/-	√	-	-	√	√
sähköposti integraatio	√	√	-	√	-	√

Tukipalvelut						
Puhelintuki	√	-	-	√	-	-
Manuaali	√	√	√	√	√	√
FAQ	√	√	√	√	√	√
Tutoriaalit	√	√	√	√	√	√
Keskustelut	√	√	√	√	√	√
Web-lomakkeet	√	-	-	√	-	-
Tuetut alustat						
Windows	√	√	√	√	√	√
Linux	√/-	√	√	√	√	√
Solaris	-	√	√	√	√	√
Unix	√/-	√	√	√	√	√
Mac OS X	-	√	√	√	√	√

## LIITE 2. Sovellusten käytettävyyssarviotaulukko.

1= täysin samaa mieltä

5= täysin eri mieltä

Sovellus	Easy Projects .NET	SugarCRM	tinyERP	Trac	ActiveCollab	DotProject
1. Kokonaisuudessaan, olen tyytyväinen ohjelman helpokäyttöisyyteen.	2	2	4	2	1	4
2. Ohjelma oli yksinkertainen käyttää.	2	2	3	1	2	3
3. Pystyin nopeasti suorittamaan tehtäviä ja tehtäväkokonaisuuksia ohjelman avulla.	3	2	3	2	2	3
4. Pystyin tehokkaasti suorittamaan tehtäviä ja tehtäväkokonaisuuksia ohjelman avulla.	2	2	3	3	2	3
5. Ohjelmaa oli mukava käyttää.	4	3	3	2	2	4
6. Ohjelma oli helppo käyttää	2	2	3	3	2	3
7. Ohjelmiston virheilmoitukset olivat selkeitä ja informoivat miten virhe tulisi korjata.	2	2	4	4	3	5
8. Jos tein virheen ohjelmaa käytettäessä, pystyin korjaamaan virheeni vaivatta.	2	2	3	3	3	5
9. Ohjelmistoon liittyvän informaation löytäminen oli helppoa.	2	3	3	3	2	2
10. Ohjelmistoon liittyvä informaatio oli selkolukuis-ta ja ymmärrettävää.	3	3	3	3	2	3
11. Informaatio oli selkeästi jäsennetty omiin kategorioihinsa.	2	2	3	2	2	4
12. Ohjelman käyttöliittymä oli miellyttävä.	3	3	4	2	2	4
13. Tykkäsin käyttää ohjelman käyttöliittymää.	3	3	4	2	2	5
14. Ohjelma pitää sisällään kaikki ominaisuudet joita odotin sen sisältävän.	1	1	2	3	1	2

15. Kokonaisuudessaan pidin ohjelmistosta.	2	2	4	2	1	3
<b>Keskiarvo</b>	<b>2,3</b>	<b>2,3</b>	<b>3,3</b>	<b>2,5</b>	<b>1,9</b>	<b>3,5</b>