

SHAMAN-PROJEKTI

Marko Andersson

Mika Rinkinen

Matti Törmä

Timo Valonen

Projektisuunnitelma

1.4.2005

Versio 1.0

Jyväskylän yliopisto

Tietotekniikan laitos

Tietoja projektista ja dokumentista

Tekijät:	Marko Andersson (maenande@cc.jyu.fi) Mika Rinkinen (mikarin@cc.jyu.fi) Matti Törmä (matorma@cc.jyu.fi) Timo Valonen (tjjovalo@cc.jyu.fi)
Yhteystiedot:	Agora, huone AgC223.3. Puhelin (014) 260 4965. Postilista shaman@korppi.jyu.fi. Postilistan arkisto https://korppi.it.jyu.fi/list-archive/shaman/ . WWW-sivu http://sovellusprojektit.it.jyu.fi/shaman .
Työn nimi:	SHAMAN-projekti, projektisuunnitelma
Työ:	Projektisuunnitelma tietotekniikan sovellusprojektiin
Sivumäärä:	28
Tilaaaja:	Jyväskylän yliopisto, ATK-keskus
Tiivistelmä:	SHAMAN-projekti suunnittelee keskitetystä käyttäjänhallinnasta vastaavan AMAN-järjestelmän seuraajan. Projekti toteuttaa järjestelmän prototyyppiasteelle. Dokumentti kuvaa projektin suunnitellun läpiviennin.
Avainsanat:	AMAN, tietokanta, käyttäjätunnus, salasana, käyttäjähallinta, hallinta.

Versiohistoria

Versio	Päiväys	Tekijä	Kuvaus
0.203	3.2.2005	Marko Andersson	Luvut 1 ja 2.
0.207	7.2.2005	Marko Andersson	Luku 6.
0.208	8.2.2005	Marko Andersson	Luku 3.
0.216	16.2.2005	Marko Andersson	Luvut 4 ja 5.
0.217	17.2.2005	Marko Andersson	Luvut 4 ja 5.
0.218	18.2.2005	Marko Andersson	Johdanto. Luvut 2 ja 8.
0.221	21.2.2005	Marko Andersson	Luku 7.
0.222	22.2.2005	Marko Andersson	Luvut 8.5, 3.2.
0.228	28.2.2005	Marko Andersson	Kirjoitusasun korjausta, sivut I-IV.
0.301	1.3.2005	Marko Andersson	Kirjoitusasun korjausta, Luvut 1-2.
0.302	2.3.2005	Marko Andersson	Kirjoitusasun korjausta, luvut 2-4.
0.306	6.3.2005	Marko Andersson	Kirjoitusasun korjausta, luvut 4-6.
0.307	7.3.2005	Marko Andersson	Kirjoitusasun korjausta, luvut 6-9.
0.311	11.3.2005	Marko Andersson	Kirjoitusasun korjausta, luvut 1-8.
0.315	15.3.2005	Marko Andersson	Kirjoitusvirheiden korjausta, luvut 1-8.
0.317	17.3.2005	Marko Andersson	Asiavirheiden korjausta, termien lisäämistä.
0.318	18.3.2005	Marko Andersson	Kirjoitusasun korjausta, luvut 1-8.
0.321	21.3.2005	Marko Andersson	Kirjoitusasun korjausta, luvut 1-4.
0.322	22.3.2005	Marko Andersson	Kirjoitusasun korjausta, luvut 5-8.
0.329	29.3.2005	Marko Andersson	Sisällön muutosta. Luku 5.1.
1.0	31.3.2005	Marko Andersson	Ulkoasun korjausta lukuun 5.1. Julkaistava versio 1.0.

Termiluettelo

Projektin aiheeseen ja toteutustekniikoihin liittyviä termejä ja käsitteitä ovat seuraavat:

Acta	on Jyväskylän avoimessa yliopistossa käytössä oleva opintosuoritusrekisteri.
Alaryhmät	ovat ryhmiä, jotka ovat tietyn ryhmän jäseniä.
Apache	on ilmainen HTTP-palvelinohjelmisto.
SHELL-skripti	on UNIX- ja Linux-koneissa ajettava komentorivitiedosto.
CSS	(Cascade Style Sheets) on WWW-sivujen ulkoasun määrittelyyn käytetty kieli, jolla voidaan erottaa sivujen sisältö ja esitysasu toisistaan.
Fortime	on rekisteri, jossa säilytetään tietoja henkilökunnasta ja muista Jyväskylän yliopistolta palkkioita saavista.
FunetEduPerson	on skeema, joka määrittelee henkilön attribuutit korkeakoulujen välisessä viestinnässä.
HAKA	on Suomen yliopistojen ja ammattikorkeakoulujen yhteinen käyttäjätunnistusjärjestelmä. Siihen liittyneen korkeakoulun käyttäjät pääsevät yhdellä kirjautumisella korkeakoulusektorin palveluihin riippumatta, siitä kuka palvelun tuottaa.
Henkilö	kuvaava henkilöä ja hänen tietojaan.
HTML-template	on Perlin moduuli, joka mahdollistaa skriptien erottamisen HTML-sivusta.
JORE	on Jyväskylän yliopistossa kehitetty ja käytössä oleva opintosuoritusrekisteri.
Kohdejärjestelmät	ovat järjestelmiä, joihin käyttäjähallintajärjestelmä vie tietoa. Kohdejärjestelmiä käytetään rajapintojen kautta, ja niitä ovat mm. Kerberos, LDAP ja CGI.

Korppi	on Jyväskylän yliopiston opintotietojärjestelmä.
Lähdejärjestelmät	ovat järjestelmiä, joista käyttäjähallintajärjestelmä saa tietonsa käyttäjistä. Järjestelmät ovat Fortime, Jore, Acta ja Korppi sekä salasana.jyu.fi.
Perl	on ilmainen skriptikieli, jolla voidaan toteuttaa myös ajettavia ohjelmia.
PostgreSQL	on ilmainen tietokannanhallintajärjestelmä.
Ryhmä	on osana ryhmähierarkiaa ja siihen voi liittyä muita ryhmiä. Ryhmä voi myös kuvata erilaisia rooleja. Tunnuskohtaiseen ryhmään voi liittyä vain yksi tunnus.
Shibboleth	on organisaatorajat ylittävä käyttäjien hallintaprotokolla, joka tarjoaa autentikointi-, autorisointi- ja pääsynvalvontapalveluja.
Tunnukset	ovat henkilöiden käyttäjätunnuksia. Henkilöllä voi olla useampia tunnuksia.
Yläryhmät	ovat tietyn ryhmän yläpuolella olevia ryhmiä, joiden jäsenenä ryhmä on.

Sisältö

1 JOHDANTO	1
2 PROJEKTIN TAUSTAA JA TAVOITTEET	3
2.1 AMAN JA YLIOPISTON TIETOJÄRJESTELMÄT	3
2.2 AMANIN HEIKKOUSIA	5
2.3 TIETOKANNAN TAVOITTEET	6
2.4 KÄYTTÖLIITTYMÄN JA RAJAPINTOJEN TAVOITTEET	7
2.5 OPPIMISTAVOITTEET	9
3 PROJEKTIN RESURSSIT	10
3.1 PROJEKTIORGANISAATIO	10
3.2 TYÖTILA, LAITTEET JA OHJELMISTOT	11
4 PROJEKTIN TEHTÄVÄT JA NIIDEN JAKAUTUMINEN	12
4.1 VASTUUALUEJAKO	12
4.2 TEHTÄVIEN TYÖMÄÄRÄ JA JAKAUTUMINEN	12
5 PROJEKTIN AIKATAULU	14
5.1 VAIHEET JA TULOKSET	14
5.2 TEHTÄVIEN AIKATAULU	15
6 RISKIEN ARVIOINTI JA HALLINTA	17
6.1 RISKITÄULUKKO	17
6.2 HENKILÖIHIN LIITTYVÄT RISKIT	17
6.3 AIHEALUEESEEN LIITTYVÄT JA TEKNISET RISKIT	19
7 PROJEKTIN KÄYTÄNTEET	21
7.1 DOKUMENTOINTI	21
7.2 KOKOUKSET	22
7.3 TIEDOTTAMINEN	22
7.4 TIEDOSTOT JA HAKEMISTOT	23
7.5 CVS-HAKEMISTORAKENNE	25
7.6 TESTAUS	25
7.7 PEREHDYTYKSET JA KOULUTUKSET	26
8 YHTEENVETO	27
LÄHTEET	28

1 Johdanto

SHAMAN-niminen tietotekniikan sovellusprojekti suunnittelee ja toteuttaa käyttäjätunnusten ja salasanojen sekä käyttöoikeuksien hallintaan käytettävän järjestelmän. Järjestelmä kehitetään AMAN-nimisen järjestelmän korvaajaksi, joka ei enää vastaa sille asetettuja uusia vaatimuksia.

AMAN-järjestelmällä hallitaan Jyväskylän yliopiston opiskelijoiden, henkilökunnan ja muiden tahojen käyttäjätunnuksia ja salasanoja eri kohdejärjestelmissä. AMAN välittää käyttäjätunnukset ja salasanat eri autentikointipalvelimille, joista eri järjestelmät voivat autentikoida käyttäjän. AMAN saa tietonsa automaattisesti lähdejärjestelmistä, joita ovat mm. JORE, Fortime, Acta ja Korppi.

AMAN-järjestelmän heikkoutena on tietokannan puutteellinen suunnittelu, joten tietokanta ei enää suoriudu uusista vaatimuksista. Käyttöliittymä on toteutettu skripteillä, joten sen ylläpitäminen ja muokkaaminen on hyvin hankalaa.

SHAMAN-projekti määrittelee järjestelmälle asetettavat tavoitteet ja vaatimukset nykyisten ja tulevien tarpeiden pohjalta. Projekti myös määrittelee, suunnittelee ja toteuttaa tietokannan, joka vastaa uuden järjestelmän tarpeita. Ryhmä myös määrittelee, suunnittelee ja toteuttaa rajapinnat, joiden kautta järjestelmään voidaan syöttää tietoja eri lähdejärjestelmistä, sekä rajapinnat tietojen vientiin kohdejärjestelmiin.

Projektisuunnitelma kuvaa projektin läpivientiä, tavoitteita ja käytänteitä. Lisäksi ryhmä laatii sovellukselle vaatimusmäärittelyn ja sovellussuunnitelman. Vaatimusmäärittely kuvaa sovelluksen tarjoamat tiedot ja toiminnot eri käyttäjäryhmille. Vaatimusmäärittely sisältää myös järjestelmän tekniset vaatimukset sekä kuvaukset käyttötapauksista. Sovellussuunnitelma kuvaa, miten vaatimusmäärittelyssä olevat vaatimukset toteutetaan sovelluksessa. Ohjelman rakennetta kuvataan moduulien ja niiden aliohjelmien avulla sekä määritellään miten, eri osat keskustelevat keskenään ja muiden järjestelmien kanssa.

Muita dokumentteja ovat testaussuunnitelma ja testiraportti, joissa kuvataan sovelluksen testausympäristö ja testitapaukset, sekä testikertojen tulokset. Projektin lopussa suunnitelmien toteutumista arvioidaan vertaamalla suunnitteludokumentteja myöhemmin laadittaviin projekti- ja sovellusraportteihin.

Projektin taustoja ja tavoitteita kuvataan luvussa 2. Projektin organisaatiota ja resursseja käsitellään luvussa 3. Luvussa 4 käsitellään projektin tehtäviä sekä niiden työmäärää ja jakautumista jäsenten kesken. Projektin aikataulu esitetään luvussa 5. Projektin riskien arviointia ja hallintaa tarkastellaan luvussa 6. Projektin käytänteitä palaverien, pöytäkirjojen ja tiedostojen osalta kuvataan luvussa 7.

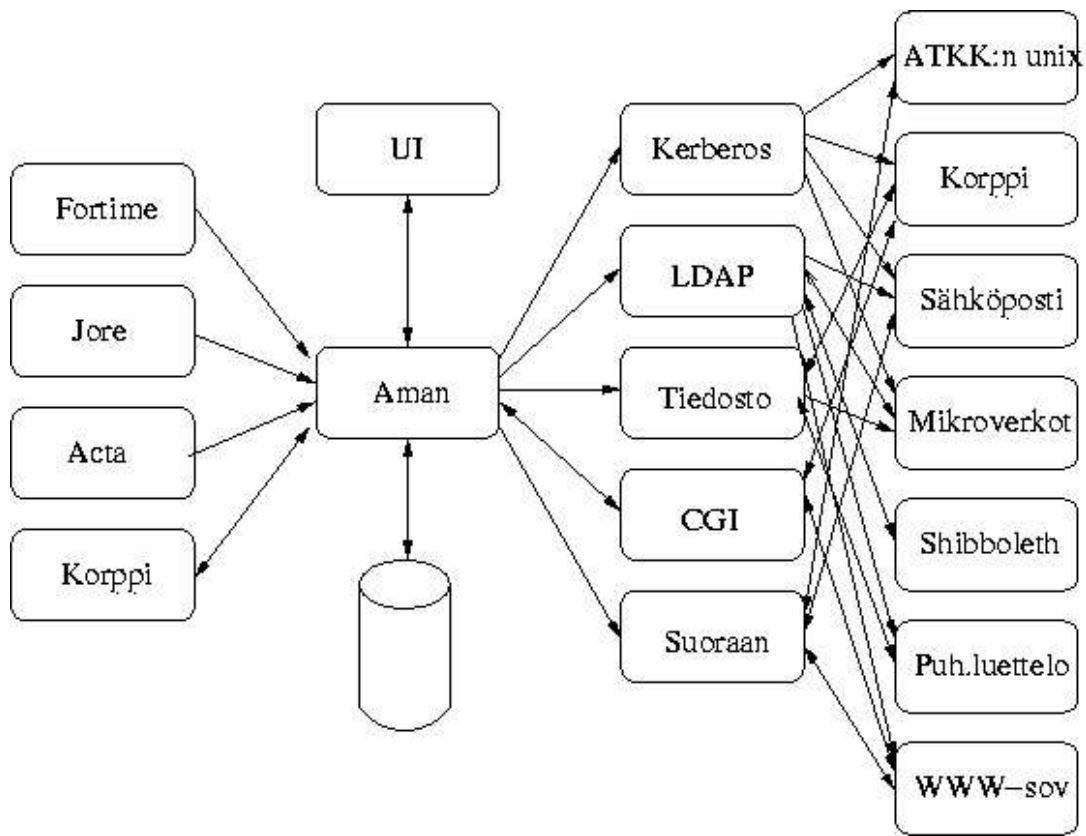
2 Projektin taustaa ja tavoitteet

Projekti suunnittelee ja toteuttaa järjestelmän, joka korvaa vanhan AMAN-järjestelmän. Uudessa järjestelmässä pyritään välttämään vanhan järjestelmän heikkouksia sekä huonoja toteutustapoja, joita ovat esimerkiksi skripteinä käytetty ohjelmointikieli sekä käyttöliittymän upottaminen osaksi skriptejä. Vaatimukset ja toteutustekniikat kuvataan tarkemmin vaatimusmäärittelyssä ja sovellussuunnitelmassa.

2.1 AMAN ja yliopiston tietojärjestelmät

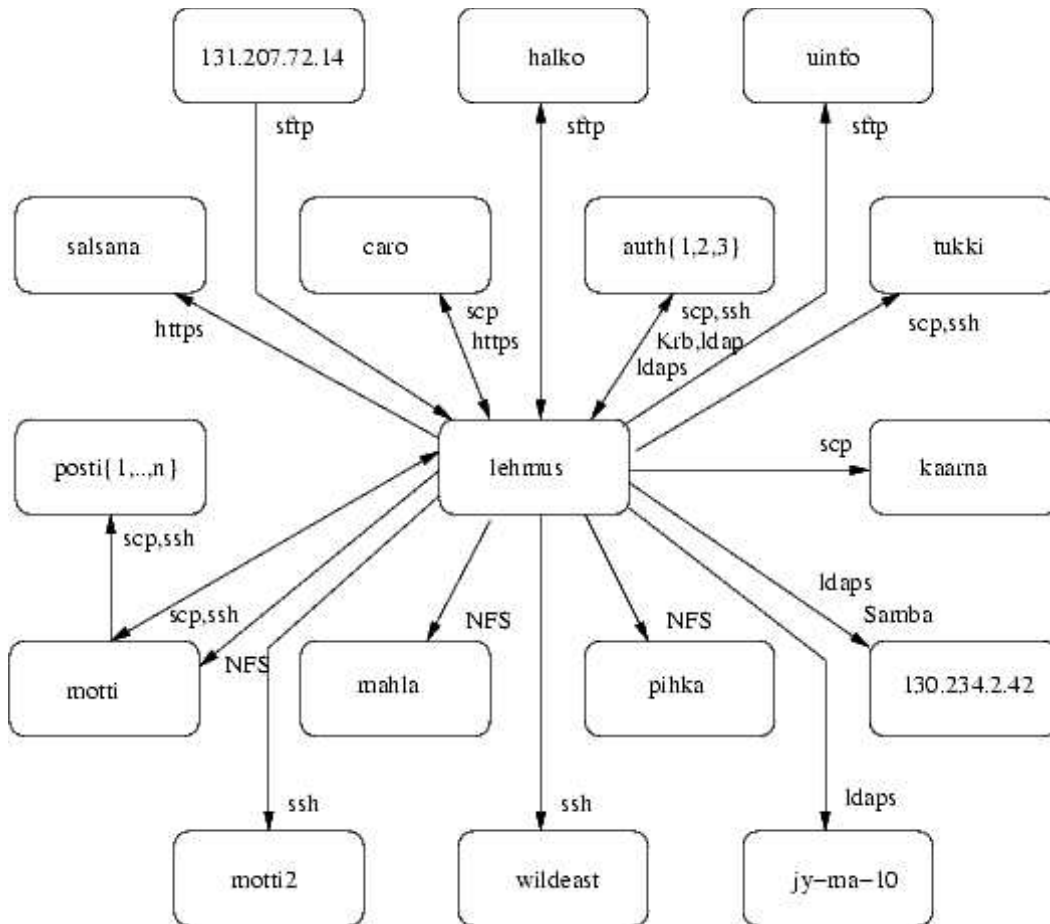
AMAN-järjestelmä on vuonna 2001 käyttöönotettu ATK-keskuksen käyttäjien ja käyttöoikeuksien hallintajärjestelmä, jonka päätehtävänä on ylläpitää Jyväskylän yliopiston eri tietojärjestelmien käyttöoikeuksia. AMAN on toteutettu Perl- ja SHELL-skripteillä sekä tiedot on tallennettu PostgreSQL-tietokannanhallintajärjestelmällä.

Kuva 1 esittää AMANille tietoja toimittavat lähdejärjestelmät sekä kohdejärjestelmät, joille AMAN välittää käyttöoikeustietoja.



Kuva 1. AMAN-järjestelmän yhteydet muihin järjestelmiin [Toivonen].

Kuvassa 2 on Jyväskylän yliopiston palvelimet, joiden kanssa AMAN-järjestelmä keskustelee.



Kuva 2. AMAN-järjestelmä on yhteydessä useisiin palvelimiin [Toivonen].

2.2 AMANin heikkouksia

Keskeisin tarve kehittää korvaaja AMAN-järjestelmälle on käyttäjien suuren vaihtuvuuden asettamat vaatimukset järjestelmän ylläpidettävyydelle. ATK-keskuksen järjestelmissä vuosittainen vaihtuvuus on n. 2000 henkilöä. Sama henkilö voi esiintyä eri järjestelmissä eri rooleissa, joita ovat mm. opiskelija tai henkilökuntaan kuuluva. Yliopistolla on käytössä useita tietojärjestelmiä, joissa jokaiseen on toteutettu erillinen käyttäjien autentikointi.

AMAN-järjestelmässä tuki saman henkilön eri rooleille eri organisaatioissa ja eri järjestelmissä on puutteellinen, sillä nykyiseen järjestelmään ei ole sisällytetty ryhmäkäsitteitä. Nykyiset ryhmät perustuvat UNIX-järjestelmästä perittyihin ryhmiin. Nykyinen käyttöoikeuksien hallintajärjestelmä soveltuu huonosti erityisesti avoimen

yliopiston opiskelijoiden sekä yliopistolla väliaikaisesti kirjoilla olevien opiskelijoiden ja henkilökuntaan kuuluvien käyttöoikeuksien hallintaan. Käyttäjätunnusta luotaessa sille ei voida määrittää voimassaoloaikaa AMAN-järjestelmässä. Tämä hankaloittaa varsinkin kurssikohtaisia oikeuksien hallintaa, jolloin oikeudet on lisättävä ja poistettava käsin.

Merkittävä heikkous AMAN-järjestelmässä on käyttöliittymä, jonka ulkoasun muokkaaminen vaatii ohjelmointitietämystä. Käyttöliittymä on koodattu osaksi Perl-koodia, jolloin sen muuttaminen on äärimmäisen hankalaa ja aikaavievää. AMAN-järjestelmä muodostuu noin 50 skriptistä.

Järjestelmää on laajennettu kasvavien vaatimusten ja tarpeiden mukaisesti sen sijaan, että järjestelmä olisi alusta alkaen suunniteltu tukemaan yliopiston tietojärjestelmien keskitettyä käyttäjähallintaa. AMAN-järjestelmää ei ole siten mieltä enää kehittää, vaan järjestelmä on toteutettava kokonaan uusiksi.

2.3 Tietokannan tavoitteet

Ryhmä määrittelee, suunnittelee ja toteuttaa tietokannan uuteen Jyväskylän yliopiston käyttäjähallintajärjestelmään. Tietokannan tulee sisältää tiedot yksittäisestä käyttäjästä ja hänen oikeuksistaan eri kohdejärjestelmissä. Tietokantaan on voitava määrittää käyttäjän postilaatikon sekä kotihakemiston sijainti.

Tilaajan vaatimuksesta yhdelle henkilölle on tarvittaessa voitava määrittää useita eri tunnuksia sekä yhdelle tunnukselle eri salasanoja eri järjestelmiin. Näin voidaan taata riittävä tietoturva eri järjestelmien välillä, vaikka yhden järjestelmän salasana paljastuisi. Tästä mallista voidaan helposti siirtyä käyttämään yhden salasanan mallia, jos siihen on tulevaisuudessa tarvetta.

Tietokantaan on myös suunniteltava ja toteutettava ryhmäkäsitteet, joilla voidaan hallita erilaisia käyttäjäryhmiä. Ryhmät muodostavat samalla organisaatiohierarkian, jonka avulla

voidaan määritellä yksittäisen henkilön kuuluminen eri organisaatioihin. Ryhmiin liitetään tiedot ryhmien oikeuksista eri järjestelmiin, jolloin kaikki ryhmään kuuluvat saavat samat oikeudet. Yksittäisillä ryhmän jäsenillä voi olla erityisoikeuksia, jotka määräytyvät jonkin muun ryhmän kautta, sillä yksi henkilö voi kuulua moneen eri ryhmään. Ryhmiin liitetään myös tiedot ryhmien sähköpostilaatikoista ja kotihakemistoista, jolloin ryhmällä voi olla yhteinen postilaatikko. Yksittäisellä käyttäjällä on oma henkilökohtainen ryhmä, johon voi kuulua vain käyttäjä itse. Tämä ryhmä perii oikeudet muista ryhmistä.

Ryhmiin kuuluu tieto ryhmän voimassaoloajasta, jolla voidaan kontrolloida esimerkiksi kurssin ajaksi annettuja oikeuksia. Ryhmille määritelty tila kertoo, onko ryhmä aktiivinen, väliaikaisesti poissa käytöstä vai lukittu. Näin voidaan esimerkiksi yksittäisiä käyttäjätunnuksia lukita lukitsemalla tunnuksen henkilökohtainen ryhmä.

Henkilöstä kerätään tietokantaan mahdollisimman yleiskäyttöistä tietoa ja pyritään mahdollisuuksien mukaan välttämään järjestelmäkohtaista tietoa. Kun tieto on tarpeeksi abstraktia, sitä voidaan hyödyntää paremmin muissa järjestelmissä, sillä rajapinnat huolehtivat tiedon muuntamisesta järjestelmien ymmärtämään muotoon. Tietojen tulee myös toteuttaa funetEduPerson-skeeman, jolloin tietoja voidaan käyttää mahdollisimman kattavasti myös tulevaisuudessa.

2.4 Käyttöliittymän ja rajapintojen tavoitteet

Rajapintojen tulee tarjota liittymät projektissa kehitettävään tietokantaan tietojen tuomiseksi, viemiseksi sekä muuttamiseksi. Tietoja tuodaan järjestelmään lähdejärjestelmien rajapinnan kautta, joka ottaa vastaan lähdejärjestelmiltä tulevat siirtotiedostot tai viestit, ja pyytää tietokannanhallintamoduulia muokkaamaan tietokantaa niiden perusteella. Tietojen siirtoa AMAN-järjestelmään tulee valvoa, sillä tiedot eivät välttämättä ole oikeita, tai sama asia on voitu ilmaista useammalla eri tavalla. Tällaisesta on esimerkkinä laitoksen nimi, joka on voitu lyhentää usealla eri tavalla. Tietokannassa tietojen tulee olla yksikäsitteisiä.

Kohdejärjestelmien rajapinta muodostaa eri kohdejärjestelmille tarjottavat tiedot. Näihin kuuluvat mm. autentikointipalvelu Kerberos ja LDAP-hakemisto. Tietojen vieminen suoritetaan pääasiassa eräajoina, mutta käyttöliittymän kautta muutetut tiedot tulee siirtää kohdejärjestelmiin välittömästi.

Kohdejärjestelmän rajapinta toimii viestinvälittäjänä useiden erilaisten, kohdejärjestelmiä käyttävien, moduulien välissä. Moduuleja vaihtamalla ja muokkaamalla voidaan reagoida kohdejärjestelmissä tapahtuviin muutoksiin.

Tietoja voidaan muokata erillisten käyttöliittymien kautta. Opiskelijoille ja pääkäyttäjille on erilliset käyttöliittymät. Tämän lisäksi muilla henkilöillä voi olla oikeus lukea tiettyjä tietoja. Näissä tapauksissa heillä on oikeudet vain tiettyihin osiin tai tietoihin pääkäyttäjän käyttöliittymässä.

Käyttöoikeuksien hallintaan ja muokkaukseen on suunniteltava automaattiset rutiinit koko tunnuksen elinkaaren ajaksi. Tunnukseen voidaan liittää päivämääriä, jolloin järjestelmä muuttaa automaattisesti tunnuksen tilaa. Näin voidaan kontrolloida tunnusten vanhenemista ja kurssikohtaisia oikeuksia, jotka menetetään kurssin loputtua. Tilamuutoksiin voidaan myös määritellä toimintoja, kuten postilaatikon poisto. Jos esimerkiksi opiskelija siirtyy opettajaksi, on hänen tilansa muututtava järjestelmässä, jolloin hän saa opettajille kuuluvia oikeuksia eri kohdejärjestelmiin tietyssä organisaatiossa.

Käyttöliittymä tulee olla helposti muokattavissa jälkikäteen, joka otetaan huomioon esimerkiksi suunniteltaessa käyttöliittymän käytettävyyttä tai käännettäessä sitä muille kielille. Tästä johtuen käyttöliittymä toteutetaan Perlin HTML-Templates -moduulilla, joka mahdollistaa edellä mainitut ominaisuudet.

2.5 Oppimistavoitteet

Projektin tavoitteena on antaa ryhmän jäsenille konkreettinen kuva laajan ohjelmistoprojektin eri vaiheista ja oikean sovelluksen kehittämisestä. Työn suunnittelun ja seurannan oppiminen auttavat projektin läpiviennissä ja aikataulussa pysymisessä. Jos työmäärä ei jakaudu tasaisesti projektijäsenten kesken, on sitä tasattava sopivasti uudelleenorganisoimalla töitä ryhmän kesken.

Projektityöskentelyn tarkoituksena on myös kehittää opiskelijoiden ryhmätyötaitoja. Ristiriitatilanteet tulee oppia sopimaan ryhmän sisällä, sillä kiristyneestä ilmapiiristä on vain haittaa projektin etenemiselle. Opiskelija joutuu myös ottamaan vastuuta sekä omasta työskentelystään, että ryhmän toimista.

Jäsenet oppivat kirjoittamaan erilaisia raportteja sekä dokumentteja, joita käytetään projektin suunnitteluun ja raportointiin. Myös oman työn raportointi nousee merkittävään asemaan.

Ryhmä saa myös kokemusta Perl-ohjelmointikielestä, joka ei ole kovinkaan tuttu ryhmälle entuudestaan. Myös tietokantojen suunnittelu tulee tutuksi projektin aikana.

3 Projektin resurssit

Luvussa kuvataan projektin organisaatiota ja käytettävissä olevia resursseja.

3.1 Projektiorganisaatio

Projektiorganisaatioon kuuluu neljä tietotekniikan opiskelijaa, jotka suunnittelevat ja toteuttavat järjestelmän tietotekniikan sovellusprojektina. Ryhmään kuuluvat Marko Andersson, Mika Rinkinen, Matti Törmä ja Timo Valonen.

Tilaaajan edustajina toimivat Matti Levänen ja Tero Toivonen Jyväskylän yliopiston ATK-keskuksesta. Projektiorganisaatioon kuuluu myös Sirpa Turjanmäki, joka vastaa nykyisen AMAN-järjestelmän toiminnasta.

Teknisenä ohjaajana toimii Lassi Paavolainen sekä vastaavana ohjaajana Jukka-Pekka Santanen tietotekniikan laitokselta.

Projektissa kuullaan myös eri tieto- ja käyttöjärjestelmien asiantuntijoita, jotka auttavat projektia kartoittamaan AMAN-järjestelmään tallennettavia tietoja ja kohdejärjestelmien tarpeita. Sähköpostijärjestelmien ylläpitäjistä mukana ovat Seppo Kallio ja Tapani Tarvainen sekä UNIX-järjestelmien osalta Tuomas Kautto ja Mika Videnoja. Mikroverkkojen asiantuntijana toimii Mika Mattila. Korppia edustavat Minna Hillebrand ja Pauli Kujala. Optiman edustajana on Markku Närhi. Yliopiston WWW-järjestelmien asiantuntijoina ovat Jenni Hytönen sekä Jussi Talaskivi. Kirjaston asiantuntijana kuullaan Risto Heikkistä.

Projektin ATK-tukena toimivat Jyväskylän yliopiston informaatioteknologian tiedekunnasta Mika Harju, Jussi Kokkonen, Jari Lepistö, Juha-Matti Sulander ja Harri Tuomi.

3.2 Työtila, laitteet ja ohjelmistot

Projektin työhuone sijaitsee Sovellusprojektin tiloissa Agoran toisessa kerroksessa huoneessa AgC223.3. Puhelinnumero huoneeseen on (014) 260 4965

Projektin käytössä on neljä PC-tietokonetta, joista kaksi on varustettu Windows XP -käyttöjärjestelmällä, ja toiset kaksi Linux Fedora Core 2 -käyttöjärjestelmällä.

Tietokoneisiin on asennettu perusohjelmistot, jotka sisältävät esimerkiksi toimisto-ohjelmat. Windows-koneet sisältävät sekä Microsoft Officen version 2003 (11.6359.6360) SP1, että OpenOfficen version 1.1.4. Linux-koneissa on vain OpenOfficen versio 1.1.3. Lisäksi yhteen Windows-koneeseen on asennettu Microsoft Projectin versio 9.0.2000.0224 sekä Microsoft Vision versio 6.0.2072. Linux-koneille on asennettu Perl-tulkeista versio 5.8.3 built for i386-linux-thread-multi sekä PostgreSQL-tietokannanhallintajärjestelmästä versio 7.4.7. Windows-koneisiin on asennettu ActivePerl versio 5.8.0 built for MSWin32-x86-multi-thread. Projektin käytössä on myös yksi yhteinen verkkotulostin sekä laitoksen kopiokone.

Projektin dokumentointi ja sovellukset sijoitetaan projektin kotisivuille osoitteeseen <http://sovellusprojektit.it.jyu.fi/shaman/>. Projektilla on käytössään sähköpostilista shaman@korppi.jyu.fi. Sähköpostilistan arkisto on nähtävissä osoitteessa <https://korppi.it.jyu.fi/list-archive/shaman/>. Arkisto on suojattu käyttäjätunnuksella ja salasanalla, jotta viestit eivät näkyisi kaikille käyttäjille.

Sähköpostilistalla olevalla Sirpa Turjanmäellä on pääsy ATK-keskuksen aman-devel-sähköpostilistalle. Hän voi siten lähettää projektille hyödyllisiä tietoja projektin sähköpostilistalle.

4 Projektin tehtävät ja niiden jakautuminen

Projektin tehtävät jaetaan ryhmän jäsenten kesken. Jokainen jäsen tutustuu jokaiseen projektin osa-alueeseen, mutta ryhmästä aina yksi keskittyy yhteen alueeseen muita tarkemmin.

4.1 Vastuualuejako

Projektin alkuvaiheessa koko ryhmä suunnittelee järjestelmää yhdessä, mutta kun yleiset vaatimukset ja määrittelyt saadaan tehtyä, niin jäsenet paneutuvat tarkemmin omiin tehtäviinsä. Alussa on tärkeää hyödyntää kaikkien suunnittelupanosta, jotta tietokannasta saadaan mahdollisimman toimiva.

Marko Andersson ja Mika Rinkinen perehtyvät muita enemmän Perl-ohjelmointiin, sillä heillä on eniten kokemusta ohjelmoinnista. Matti Törmän vastuualueena on tietokannan suunnittelu sekä käyttöliittymän suunnittelu ja toteutus. Timo Valonen vastaa tietokannan suunnittelusta ja toteutuksesta. Koska tietokannan taulurakenne valmistuu ennen muita järjestelmän osia, siirtyy Timo muihin tehtäviin toteutettuaan tietokannan.

Projektipäällikkönä toimii koko projektin ajan Marko Andersson, joka vastaa projektin suunnittelusta, hallinnasta ja tiedotuksesta.

4.2 Tehtävien työmäärä ja jakautuminen

Taulukossa 1 on esitetty arviot projektin tehtävien jakautumisesta, sekä niihin kuluvista työtunneista.

Tehtävä	Marko	Mika	Matti	Timo	Yhteensä
Projektin hallinta	113	43	47	48	251
Projektisopimus	-	-	4	4	8
Projektin suunnittelu	30	3	3	4	40
Aiheeseen ja työkaluihin tutustuminen	20	20	20	20	80
Tiedotus	15	5	5	5	30
Projektin hallinta	25	5	5	5	40
Projektin raportointi	20	5	5	5	35
Kotisivut	3	5	5	5	18
Palaverit	67	67	67	67	268
Palaverit ja niiden valmistelu	37	37	37	37	148
Työpajat	10	10	10	10	40
Pöytäkirjat	15	15	15	15	60
Perehdytykset	5	5	5	5	20
Suunnittelu	70	100	130	110	410
Vaatimusten määrittely	10	20	40	20	90
Tietokannan suunnittelu	5	10	40	40	95
Rajapintojen suunnittelu	30	40	10	10	90
Käyttöliittymän suunnittelu	5	10	30	30	75
Sovellussuunnitelma	20	20	10	10	60
Toteutus	80	100	60	60	300
Tietokannan toteutus	-	-	10	10	20
Tietokannan hallinta	40	40	10	10	100
Rajapinnat	30	50	10	10	100
Käyttöliittymä	10	10	30	30	80
Testaus ja viimeistely	25	35	35	53	148
Testauksen suunnittelu	5	-	5	10	20
Testiraporttipohja	-	-	-	8	8
Testaus	5	20	10	5	40
Asennus- ja käyttöohje	-	-	-	15	15
Tulosten viimeistely	-	10	10	10	30
Sovellusraportti	15	5	10	5	35
Oheiskurssi	55	45	50	50	200
Esittelyt ja niiden valmistelu	10	10	10	10	40
Luennot	20	20	20	20	80
Dokumenttien kirjoitusasu	25	15	20	20	80
Yhteensä	410	390	389	388	1577

Taulukko 1. Projektin tehtävät sekä niiden jakautuminen ja niihin kuluva aika.

5 Projektin aikataulu

Luvussa määritellään projektin alustava aikataulu, johon on merkitty projektin eri vaiheille varatut aikajänteet ja vastaavat tulosten tarkistuspistepäivämäärät.

5.1 Vaiheet ja tulokset

Ensimmäinen sovellusprojektin luento pidettiin 28. tammikuuta, jolloin jaettiin ryhmät sekä aiheet. Projektin tiivistetty aikataulu on esitetty taulukossa 2.

Aika	Vaiheet ja tulokset
29.1. – 21.3.	Aiheeseen ja työkaluihin tutustuminen.
7.2. – 12.3.	Projektin suunnittelu ja vaatimusten määrittely.
	Alustava projektisuunnitelma valmis 1.3.
	Alustava vaatimusmäärittely valmis 14.3.
14.3. – 2.4.	Sovelluksen suunnittelu.
	Alustava sovellussuunnitelma valmis 1.4.
27.3. – 5.5.	Sovelluksen toteutus.
	Testattava prototyyppi 29.4.
8.4. – 5.5.	Sovelluksen järjestelmätestausta.
	Testaussuunnitelma 18.4.
2.5. – 20.5.	Sovelluksen viimeistely.
	Testiraportit 6.5.
	Sovelluksen toteutus valmis 16. toukokuuta.

Taulukko 2. Projektin viikkoaikataulu.

Ensimmäinen projektipalaveri projektiryhmän, tilaajien edustajien ja ohjaajien kanssa järjestettiin tiistaina 1.2., jolloin tutustuttiin projektiin ja sen toteutustapoihin.

Projektin suunnittelun yhteydessä tutustutaan projektin tavoitteisiin ja vaatimuksiin ja laaditaan suunnitelma projektin läpiviemiseksi. Vaatimusten määrittelyssä tutustutaan projektin taustoihin sekä vaatimuksiin, jotka sovelluksen tulee toteuttaa. Vaiheiden tuloksina ovat projektisuunnitelma ja vaatimusmäärittely.

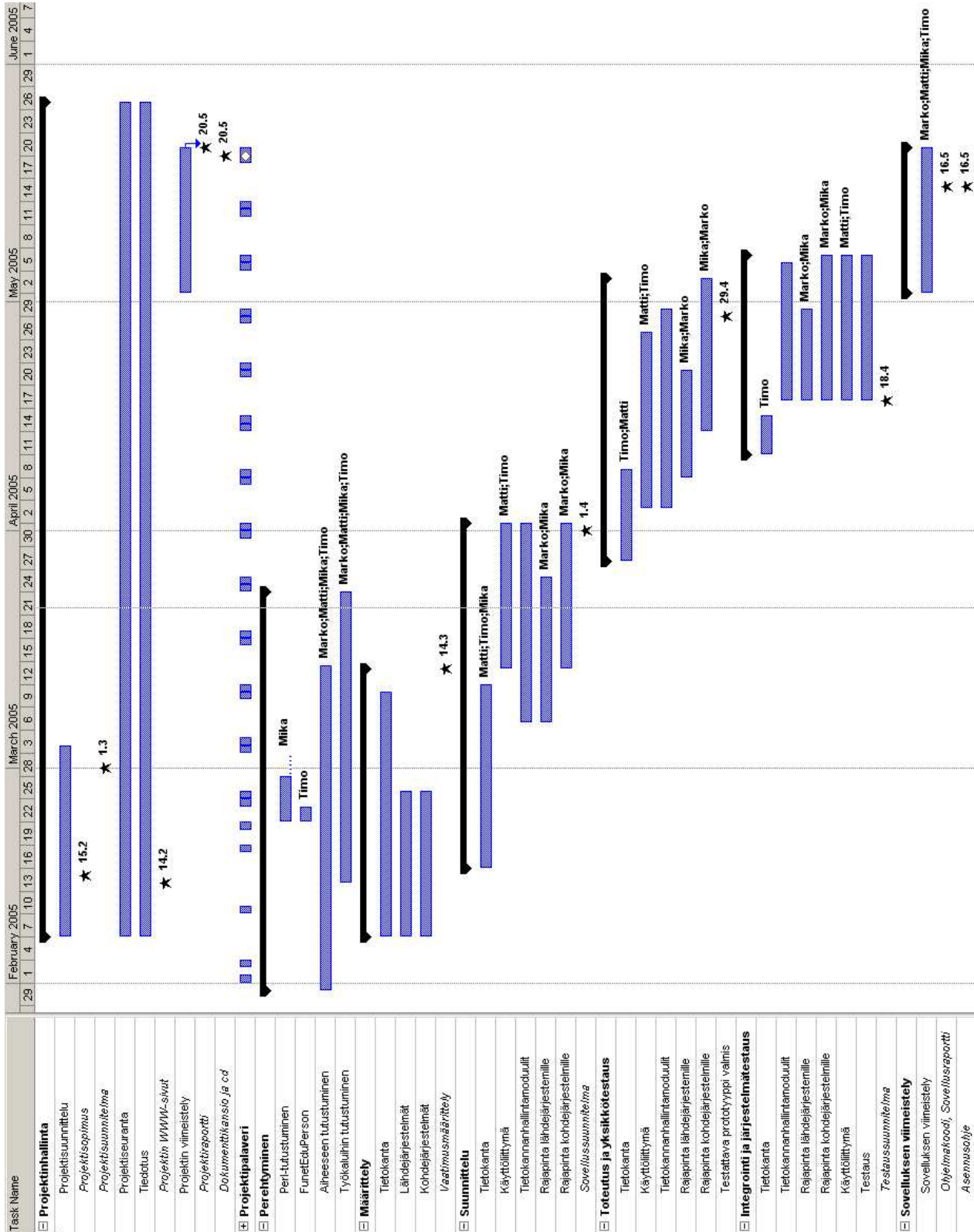
Sovelluksen suunnitteluvaiheessa suunnitellaan sovelluksen toimintoja ja rakennetta käyttötapausten pohjalta. Tietokanta pyritään toteuttamaan jo sovelluksen suunnittelun lopussa, jolloin toteutusvaiheessa voidaan käyttää valmista tietokantaa. Vaiheen tuloksena on sovellussuunnitelma.

Koska projekti on enemmän suunnittelupainoinen, toteutus aloitetaan vasta viikolla 14, jolloin kaikki määrittelyt ja suunnitelmat on saatu päätökseen. Tietokantamoduulien, käyttöliittymien ja rajapintojen toteutusta jatketaan koko huhtikuun ajan. Toteutuksen yhteydessä toteutetut ohjelmaosiot yksikkötestataan. Moduulit integroidaan järjestelmään, kun ne saadaan valmiiksi. Lopuksi koko järjestelmää testataan mahdollisuuksien mukaan.

Käyttöliittymien ja rajapintojen prototyyppejä esitellään tilaajan edustajille niiden valmistuttua huhtikuun lopussa. Tietokantaa ja sen hallintamoduulia voidaan esitellä niiden valmistuttua huhtikuun puolessa välissä. Projektin tulisi olla sovelluksen osalta valmis 7.5. mennessä, jolloin jää reilu viikko liikkumavaraa ennen projektin loppuesitystä. Dokumentoinnin viimeistelyyn saattaa kulua koko toukokuu.

5.2 Tehtävien aikataulu

Kuvassa 3 on esitetty projektin aikataulu Gantt-kaaviona. Tilanpuutteen takia kaaviosta puuttuu taulukossa 1 esitetyt oheiskurssin tehtävät.



Kuva 3. Gantt-kaavio tehtävien aikataulusta.

6 Riskien arviointi ja hallinta

Luvussa käsitellään projektiin liittyviä riskejä, niiden ennakointia, havainnointia ja ehkäisemistä. Luvussa määritellään myös toimenpiteitä, jos jokin riskeistä toteutuu.

6.1 Riskitaulukko

Taulukossa 3 on esitetty projektiin liittyviä riskejä, niiden toteutumistodennäköisyydet ja vaikutukset projektiin. Riskit on jaettu toteutumistodennäköisyyden ja vaikutuksen osalta luokkiin ei lainkaan, pieni, keskinkertainen ja suuri.

Riski	Todennäköisyys	Vaikutus
Henkilöihin liittyvät riskit		
Jäsenten kokemattomuus	Keskinkertainen	Keskinkertainen
Sairastumiset ja muut menot	Keskinkertainen	Pieni
Tiedotuksen ja viestinnän riskit	Keskinkertainen	Keskinkertainen
Ohjaajien ja tilaajan edustajien kiireet	Keskinkertainen	Pieni
Projektiryhmän koon pienentyminen	Pieni	Suuri
Aihealueen ja tekniset riskit		
Aihealueen laajuus	Keskinkertainen	Keskinkertainen
Nykyisen järjestelmän heikkouksia siirtyy uuteen järjestelmään	Pieni	Suuri
Tietoturvariskit	Keskinkertainen	Suuri
Tietoturvariskit kehitysvaiheessa	Ei lainkaan	Ei lainkaan
Toteutustyökaluista johtuvat riskit	Pieni	Keskinkertainen

Taulukko 3. Projektin riskit ja niiden toteutumistodennäköisyys.

6.2 Henkilöihin liittyvät riskit

Projektin eräänä riskitekijänä on jäsenten vähäinen kokemus projektityöstä, joka heijastuu projektin- ja ajankäytönhallinnan vaikeutena. Projektin aikataulu on suunniteltava huolellisesti, jotta välttyttäisiin mahdollisilta myöhästymisiltä. Eri tehtäviin kuluva aika on

hankala arvioida, ja tästä johtuen aikataulu tulee elämään. Tavoitteena on kuitenkin, että projekti saadaan päätökseen sovitussa aikataulussa. Projektipäällikkö voi ennakoida ja estää riskin syntymistä projektin tilan tarkkailulla. Myös ryhmän jäsenten on otettava vastuu käyttämästään ajasta sekä raportoida projektipäällikölle töiden ja annettujen tehtävien valmistumisesta. Riskin toteutuminen on todennäköistä ainakin joiltakin osin.

Ryhmän jäsenten sairastumiset tai muut menot voivat viivästyttää projektia, tai kasvattaa muiden ryhmän jäsenten työtaakkaa. Sairastumisia ei voida välttää ja niiden sattuessa on parempi, ettei sairastunut tule työskentelemään projektin tiloihin ja altistamaan muita sairastumiselle. Talvisin liikkeellä on useita eri influenssa-aaltoja, joten sairastumiset ovat todennäköisiä. Ryhmän jäsenten menot eivät pääsääntöisesti haittaa, jos niistä on ilmoitettu etukäteen. Päivän poissaolot esimerkiksi tentin takia eivät haittaa. Sen sijaan pitempiaikaiset menot, saattavat aiheuttaa myöhästymisiä aikataulussa.

Tiedotuksessa tai viestinkulussa tapahtuvat ongelmat voivat aiheuttaa turhaa työtä, joka heijastuu projektin aikatauluun. Projektin sisäinen viestintä pyritään hoitamaan siten, että kaikki ovat tietoisia tehtävistään. Jos ryhmän jäsenellä on epäselvyyksiä jonkin asian suhteen, on hänen kysyttävä ohjeita ensin ryhmän sisältä ja tämän jälkeen muilta projektiorganisaatioon kuuluvilta henkilöiltä. Näin vältetään jonkin ongelman ratkomiseen kulutettu turha työaika. Sähköposteista lähetetään aina kopiot koko ryhmälle, jolloin jokainen on perillä tulleista tai lähetetyistä sähköposteista. Sähköpostit tallennetaan arkistoon, jolloin ne eivät vahingossa tuhoudu. Viestinnän ongelmista johtuvien riskien todennäköisyys on pieni, mutta toteutuessaan ne saattavat aiheuttaa suurtakin vahinkoa.

Ohjaajien ja tilaajan edustajien kiireet eivät vaikuta suuresti projektiin. Joissakin tapauksissa saattaa olla tarpeellista saada vastaus johonkin kysymykseen nopeasti, mutta asia voidaan yleensä jättää odottamaan, kunnes siihen saadaan vastaus. Kysymykset voidaan yleensä ohjata myös muille aihetta tunteville henkilöille, joilta mahdollisesti saadaan nopeammin vastaus. Riskin toteutuminen ja vaikutus projektiin on pieni

Riskin muodostaa myös mahdollisuus, että ryhmän koko voi koska tahansa pienentyä yhdellä henkilöllä ko. henkilön elämäntilanteen johdosta. Tähän riskiin on hyvin hankala varautua, eikä sen sattuessa ole paljonkaan tehtävissä. Uuden henkilön saaminen projektiin on lähes mahdotonta. Tämän riskin toteutuminen on kuitenkin melko epätodennäköistä.

6.3 Aihealueeseen liittyvät ja tekniset riskit

Aihealueen laajuuden ja neljän kuukauden tiukan aikataulun takia projektin viivästyminen voi olla mahdollista. Tästä johtuen projektin tavoitteita sovelluksen osalta pyritään rajaamaan tarkasti ja keskittymään vain oleellisiin järjestelmän ominaisuuksiin. Myös erilaiset viime hetken vaatimukset ja tarkennukset vaikuttavat tavoitteisiin ja aikatauluun, joten näitä yritetään ennakoida mahdollisimman hyvin jo suunnitteluvaiheessa. Riskiä hallitaan tarkalla aiheen rajauksella ja tarvittaessa vaatimusten uudelleenpriorisoinnilla. Tässä yhteydessä projektin ohjaajien kokemus on korvaamatonta, sillä he pystyvät aikaisempien projektien perusteella arvioimaan projektin mahdollisuuksia toteuttaa halutut vaatimukset. Riskin toteutuminen ei ole todennäköistä, jos riskiä tarkkaillaan ja ylimääräiset vaatimukset sovitaan jatkokehitykseen. Jos riski toteutuu, on aikataulun pitämiseksi karsittava vähemmän tärkeitä toimintoja.

Koska projektiryhmä suunnittelee uudestaan jo olemassaolevan järjestelmän, on varottava, ettei nykyisen järjestelmän heikkouksia tai huonoja toteutustapoja siirretä uuteen järjestelmään. Tästä johtuen projektiryhmälle ei toimitettu vanhan järjestelmän tietokantaa heti projektin alussa, vaan tietokanta suunnitellaan alusta asti puhtaalta pöydältä. Riskiä voidaan pienentää perehtymällä olemassaolevan järjestelmän heikkouksiin sekä välttämällä niitä. Myös suunnittelulla ja eri järjestelmien ylläpitäjien mielipiteiden huomioimisella on suuri merkitys, sillä niiden kautta saadaan realistinen kuva sekä järjestelmän nykyisistä vaatimuksista että tulevaisuuden tarpeista. Riski on aika vähäinen, sillä varsinkin tilaajan edustajista Tero Toivosella on kokemusta tietokantojen ja eri järjestelmien suunnittelusta ja toteutuksesta, joten mahdolliset virheet havaitaan varmasti ajoissa.

Sovelluksen tietoturva on myös eräs riskeistä, sillä tietokannassa säilytettävät käyttäjätunnukset, salasanat sekä henkilötiedot eivät saa joutua ulkopuolisten käsiin. Sovelluksen on kyettävä käsittelemään käyttäjätunnuksia ilman, että kolmannella osapuolella on mahdollisuus saada käyttäjätunnuksia selville. Käyttäjien virheet eivät myöskään saa johtaa sovelluksen virhetoimintaan ja näin altistaa arkaluontoisia tietoja joutumasta ulkopuolisille. Riski tietojen vuotamiseen projektin aikana on lähes olematon, sillä järjestelmää testataan itse keksityllä datalla. Sitä vastoin on varottava, ettei valmiiseen järjestelmään jää vikoja, jotka altistavat tietojen vuotamiseen ulkopuolisille. Tätä voidaan vähentää testaamalla kattavasti kriittisiä järjestelmän osia.

Sovelluskehityksessä käytettävät ohjelmistot ja kieli ovat myös outoja projektin jäsenille. Vain yhdellä ryhmän jäsenistä on hieman kokemusta Perl-ohjelmoinnista, joten uuden ohjelmointikielen opetteluun ja totutteluun kuuluu hieman aikaa. Uudella kielellä ohjelmoitaessa voidaan sortua huonoihin toteutustapoihin, jotka käyttävät tarpeettomasti resursseja johonkin toiseen tapaan verrattuna tai hankaloittavat ylläpitoa. Tämä riski on kohtalaisen alhainen, sillä tekninen ohjaaja ja tilaajan edustajat osaavat puuttua ajoissa koodissa huomaamiinsa virheisiin. Perl ei myöskään sisällä ominaisuuksia, joilla voidaan helposti tehdä tietoturvan kannalta haavoittuvaa koodia, kuten puskurien ylivuodot.

7 Projektin käytänteet

Luvussa tarkastellaan projektiin liittyviä käytänteitä, kuten dokumentointia, kokouksia, tiedottamista sekä tiedon tallentamista.

7.1 Dokumentointi

Projektin dokumentointia toteutetaan yhtä aikaa muiden tehtävien kanssa, jolloin dokumentointi ei jää hatarien muistikuvien varaan projektin loppuun. Kaikki dokumentit laaditaan samoilla dokumenttipohjilla.

Esityslistoissa, pöytäkirjoissa ja tiedotteissa käytetään valmiiksi laadittua pohjaa, jossa näkyy projektin nimi ja päivämäärä sivun ylätunnisteessa. Muissa dokumenteissa (kuten projektisuunnitelmassa) käytetään valmiiksi luotua tietotekniikan laitoksen pro gradu -dokumenttipohjaa.

Dokumentit laaditaan käyttäen OpenOffice-tiedostomuotoa. Dokumenttien valmistuttua ne lähetetään ASCII-muodossa sähköpostilla tai niiden URL-osoite ilmoitetaan sähköpostilla. Tämän jälkeen dokumentit sijoitetaan projektin sivuille OpenOffice- ja PDF-muodossa. Dokumentteihin tehtyjen pienten korjausten jälkeen uusi versio siirretään WWW-sivuille, ja siitä ilmoitetaan projektin sähköpostilistalle. Suurempien asiamuutosten jälkeen muutoksista ilmoitetaan sähköpostilistalla ja korjattu dokumentti päivitetään projektin WWW-sivuille.

Lähdekoodit kommentoidaan ja nimetään käyttäen yhteisesti sovittuja käytänteitä. Lähdekooditiedostojen alkuun sijoitetaan tarkka kuvaus mm. tiedoston sisällöstä, tekijästä ja muokkauksista. Lähdekoodissa olevien funktioiden toiminnot, syötteet sekä tulosteet kuvataan funktioiden yhteydessä. Vaikeasti ymmärrettävät rivit tai toiminnot kommentoidaan erikseen. Kommentointi- ja nimeämiskielenä käytetään englantia. Lähdekoodien nimeämis- ja kommentointikäytänteet kuvataan tarkemmin sovellussuunnitelmassa

7.2 Kokoukset

Projektin viikkopalaverit pyritään pitämään joka viikko samaan aikaan. Toteutusvaiheessa voidaan muutama palavereista jättää väliin, ellei palaverissa ole mitään konkreettista käsiteltävää. Tällöin projektin tilannekatsaus lähetetään sähköpostin välityksellä. Palaverien aluksi käydään aina läpi edellisen palaverin pöytäkirja, jonka lisäksi projektipäällikkö esittää projektin tilaa käsittelevän katsauksen ja ryhmä esittää ajankäyttöraportit.

Palaverien esityslistat lähetetään osanottajille vähintään vuorokautta ennen palaverin alkua. Vähemmän virallisista tapaamisista, kuten työpajoista, ei laadita esityslistaa.

Palaverien puheenjohtajuutta kierrätetään ryhmän jäsenten kesken siten, että jokainen on vuorollaan kaksi kertaa peräkkäin puheenjohtajana. Projektin loppuvaiheessa peräkkäisten puheenjohtajan tehtävien määrä voidaan pudottaa yhteen, jolloin saadaan tasattua puheenjohtajuutta kaikkien kesken. Työpajat ja muut epäviralliset palaverit eivät kuulu kierrätyksen piiriin, joten edellisen palaverin puheenjohtaja johtaa myös työpajassa.

Palaverit pyritään nauhoittamaan myöhempää käyttöä varten. Palaverin alussa puheenjohtaja varmistaa palaveriin osallistujilta, ettei heillä ole mitään nauhoitusta vastaan. Palavereissa toimii pääsääntöisesti kaksi sihteeriä, jolloin käsiteltävät asiat saadaan paremmin muistiin. Palaverien jälkeen pöytäkirjan kirjoituksen hoitaa toinen sihteereistä.

7.3 Tiedottaminen

Projektin jäsenten välinen tiedotus hoituu pääasiassa suullisesti, sillä jäsenet näkevät toisiaan päivittäin. Työajan ulkopuolinen tiedottaminen hoidetaan ryhmän sisäisellä sähköpostilistalla, jonka ryhmän jäsenet ovat luoneet. Kiireelliset asiat voidaan hoitaa puhelimitse. Ryhmälle on luotu myös Korppiin ryhmä, jonka kautta voidaan käyttää yhteistä kalenteria. Näin mm. yhteisten palaveriaikojen sopiminen helpottuu. Korpin ryhmään on myös liitetty Lassi Paavolainen, jolloin hänenkin menonsa saadaan ryhmän tietoon.

Tilaaajan edustajille, ohjaajille ja muulle projektin organisaatiolle pyritään tiedottamaan ratkaisuksista ja ideoista niin nopeasti kuin mahdollista. Tällöin tilaajat voivat heti kommentoida ryhmän ideoita ja esittää vaihtoehtoisia menettelytapoja. Tiedottamiseen käytetään projektin sähköpostilistaa, joka on mainittu luvussa 3.2. Projektin viikkopalaverit muodostavat myös hyvän tiedotuskanavan. Ryhmä pyrkii lähettämään käsiteltävät ideat ennen palaverin alkua, jolloin tilaaajan edustajat ja ohjaajat voivat valmistautua esittämään mielipiteitä ja kommentteja ideoista.

7.4 Tiedostot ja hakemistot

Projektilla on käytössään kaksi yhteistä verkkolevyä. Niiden hakemistot nimetään yhdenmukaisesti ja kuvaavasti, jolloin jokainen ryhmän jäsen löytää helposti haluamansa dokumentit. Linux-koneissa on luotu hakemistot shaman ja shamanwww projektin hakemistoille, kun Windows-koneissa shaman -hakemisto on S- asemalla.

Shaman-hakemistoissa käytetään dokumenttien ja muiden tiedostojen luokittelussa seuraavaa hakemistorakennetta:

Ajankaytto	sisältää ajankäyttöraportit ja -kaaviot.
Dokumentit	sisältää valmiit dokumentit. Hakemiston alta löytyvät hakemistot kullekin dokumenttityypille.
Esitelmat	sisältää esitelmiin liittyvän materiaalin. Jokaisen esitelmän materiaali sijoitetaan omaan alihakemistoonsa.
Lahdekoodi	sisältää sovelluksen lähdekoodit.
Muuta	sisältää muuta materiaalia, kuten SHAMAN-logon.
Ohjeet	sisältää sovelluksen asennus- ja käyttöohjeet.
Palaverit	sisältää palaverien ja työpajojen esityslistat ja pöytäkirjat.
Sähköpostit	sisältää projektin sähköpostilistalle lähetetyt sähköpostit.
Tallenteet	sisältää palavereissa nauhoitetut tallenteet. Hakemiston alta löytyvät palaverien päivämäärän mukaan nimetyt hakemistot.

Tulosta	sisältää Linuxista tulostetut dokumentit. Hakemistoon on luotu skriptit tulosta ja pdf, jotka hoitavat tulostuksen joko kirjoittimelle tai pdf-tiedostoksi.
Vaiheessa	sisältää keskeneräiset työt. Hakemiston alta löytyy hakemistot eri tyyppisille dokumenteille sekä jokaiselle ryhmän jäsenelle oma kansio väliaikaisten tiedostojen tallentamiseen.

Projektin tulosten hakemistorakenne on seuraava:

```

./Ajankaytto
./Dokumentit/Mallit
./Dokumentit/Sopimukset
./Dokumentit/Projektisuunnitelma
./Dokumentit/Vaatimusmäärittely
./Dokumentit/Sovellussuunnitelma
./Dokumentit/Projektiraportti
./Dokumentit/Sovellusraportti
./Esitelmat
./Lahdekoodi
./Muuta
./Ohjeet
./Palaverit/Esityslistat
./Palaverit/Poytakirjat
./Tallenteet
./Tulosta
./Tulosta/out
./Vaiheessa/Marko
./Vaiheessa/Matti
./Vaiheessa/Mika
./Vaiheessa/Timo
./Vaiheessa/Palaverit
./Vaiheessa/Projektisuunnitelma

```


Tiedostot pyritään nimeämään mahdollisimman selkeästi. Tiedoston nimessä on ilmoitettava tapahtuman päivämäärä tai versio, kuten 2_workshop_21.2.2005. Päivämäärä kuvaa palaverien ja työpajojen pitopäivää, muilla dokumenteilla pääsääntöisesti muokauspäivää. Tärkeistä dokumenteista tehdään kopio omaan hakemistoonsa aina muokattaessa, joten dokumentin muokkaushistoria jää levyille.

7.5 CVS-hakemistorakenne

Projektiryhmä käyttää lähdekoodien ja dokumenttien versioiden hallintaa CVS-järjestelmää, joten ko. tiedostoista löytyy koko versiohistoria ja muutokset CVS-palvelimelta. CVS myös ennaltaehkäisee dokumentteja muokatessa sattuvia virheitä. Dokumenteista tallennetaan vain muokattavat versiot, esimerkiksi OpenOffice-muodossa olevat tiedostot.

CVS-hakemistorakenne noudattaa seuraavaa mallia:

```
dokumentit/projektisuunnitelma
dokumentit/vaatimusmaarittely
dokumentit/sovellussuunnitelma
dokumentit/esityslislat
dokumentit/poytakirjat
dokumentit/sopimukset
koodi/kayttoliittyma
koodi/rajapinnat
koodi/tietokanta
```

7.6 Testaus

Toteutusvaiheessa moduulit ja niiden yksittäiset funktiot ja aliohjelmat pyritään testaamaan mahdollisimman kattavasti, jotta suurimmat virheet huomataan. Virheiden etsimisessä voidaan lähdekoodi tarkistuttaa muilla ryhmän jäsenillä. He lukevat koodin läpi ja etsivät mahdollisia virhekohtia, joihin pitää kiinnittää testauksessa huomiota.

Integrointi- ja järjestelmätestausta suoritetaan sovelluksen toteutusvaiheessa. Kun moduuli saadaan valmiiksi, testataan sen yhteensopivuus muihin moduuleihin saman tien. Koska

kriittiset tietokantamoduulit toteutetaan ensin, on niiden toimittava virheettömästi toistensa kanssa. Testaus- ja viimeistelyvaiheessa testataan moduulien yhteensopivuus käyttämällä testauksessa virheellistä syötettä, joka poikkeaa tyypiltään odotetusta syötteestä tai syöte puuttuu kokonaan.

Lähdejärjestelmien rajapintaa testataan viemällä siihen generoituja siirtotiedostoja, jotka sisältävät keksittyä dataa. Siirtotiedostoon generoidaan myös virheitä, kuten samoja tai melkein samoja henkilöitä, jotka sovelluksen tulisi tunnistaa samaksi henkilöksi. Tero Toivonen testaa järjestelmän toimintaa myöhemmin oikealla datalla.

Kohdejärjestelmien testaus määritellään tarkemmin testaussuunnitelmassa, sillä rajapintojen toimintaa ei ole vielä määritelty.

Käyttöliittymän testauksessa käytetään eri selaimia ja niiden versiota. Käyttöliittymän tulee toimia samalla tavalla jokaisessa selaimessa. Selainkohtaisia ratkaisuja ei siis saa ilmetä, vaikka valtaosa käyttäjistä käyttäisikin jotakin tiettyä selainta.

7.7 Perehdytykset ja koulutukset

Ryhmän jäsenet saivat koulutusta Perl-ohjelmointikielestä osana sovellusprojektin oheiskurssia. Perehdytys oli hyvin pintapuolinen ja kattoi vain Perl-kielen tärkeimmät rakenteet ja ominaisuudet. Ryhmä saa myös koulutusta CVS-järjestelmästä, projektin hallinnasta ja johtamisesta, tekijänoikeuksista sekä käytettävyydestä.

Ryhmä tutuistui myös Tero Toivosen opastuksella AMAN-järjestelmän ylläpitäjän käyttöliittymään ja sen toimintoihin.

8 Yhteenveto

Projektin vaativin tehtävä on saada kattava kuva AMAN-järjestelmästä ja siihen liittyvistä lähde- ja kohdejärjestelmistä sekä tiedoista, joita uuteen käyttöoikeuksien hallintajärjestelmään tullaan sisällyttämään. Tästä johtuen projekti on suunnittelupainotteinen, josta ryhmän jäsenillä on vähän kokemusta. Jäsenet ovat aiemmin tottuneet tekemään vain ennalta määritettyjä projekteja ja erilaisia ohjelmointitöitä. Vaatimusmäärittelyjen ja suunnitelmien tekoon onkin tämän johdosta varattu pääosa projektin ajasta. Toteutus on hyvin rajattu ja sisältää vain prototyyppiasteelle kehitetyn käyttöliittymän, jota tulee olla helppo jatkokehittää.

Projektin läpivientiin ja tuloksiin vaikuttaa monia erilaisia tekijöitä, joihin jäsenet ovat yrittäneet varautua mahdollisimman hyvin. Erilaiset aikataulu- tai totutusongelmat voivat viivyttää projektia. Projektissa joudutaan suunnittelemaan toimintoja, joiden soveltuvuudesta tulevan järjestelmän tarpeisiin ei ole tarkkaa tietoa.

Lähteet

Kettunen Sami, ”Onnistu Projektissa”, WS Bookwell Oy, Juva, 2003.

Santanen Jukka-Pekka, ”Opinnäytteiden kirjoittaminen, lyhyt oppimäärä”, saatavilla HTML-muodossa osoitteesta <URL:

<http://www.mit.jyu.fi/santanen/info/kirjoittamisesta.html> >, 23.8.2000.

Santanen Jukka-Pekka, ”Tietotekniikan Sovellusprojektien ohje”, saatavilla HTML-muodossa osoitteesta <URL:

<http://www.mit.jyu.fi/palvelut/sovellusprojektit/projohje.html> >, 27.1.2005.

Toivonen Tero, ”AMAN-järjestelmän kaaviot”, saatavilla PNG-muodossa osoitteesta <URL:

<http://www.cc.jyu.fi/~tptoivon/aman/> >, 2.3.2005.