

Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

Tietokantajärjestelmät

2010

Petri Havia

Integraatiokehitysprojekti kahden tietokantajärjestelmän välillä



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma | Tietokantajärjestelmät

Kesäkuu 2010 | 36

Minna-Kristiina Paakki

Petri Havia

Integraatiokehitysprojekti kahden tietokantajärjestelmän välillä

Tämän opinnäytetyön tarkoituksena on kuvata integraatiokehitysprojektin kulku. Kyseisen projektin tavoitteena oli suunnitella, kehittää ja käyttöönottaa automatisoitu integraatio kahden yrityksen tuotekehitysyksikössä käytössä olevan tietokantajärjestelmän välille. Tiedon siirtotarve kohdistui aikataulutietoihin joita ylläpidettiin molemmissa järjestelmissä ja tämän tiedon synkronointi nähtiin kriittiseksi ajantasaisen näkyvyyden ja tiedon laadun kannalta.

Tarvemäärittelyn jälkeen hanke projektoitiin tuotekehityksen ja tietohallinnon yhteiseksi hankkeeksi ja toteutettiin pääsääntöisesti yrityksen sisäisin resurssein. Oma osuuteni projektista sisälsi tarvemäärittelyn, konseptin kehittämisen, testaamisen ja käyttäjien kouluttamisen. Työssä käytimme yrityksen käytössä olevia prosessi- ja tietorakenteiden mallinnustyökaluja.

Kahden erilaiselle alustalle rakennetun järjestelmän yhteensovittaminen oli haastava, mutta palkitseva hanke ja se toteutettiin yrityksen standardien ja arkkitehtuurillisten ohjenuorien mukaisesti. Lopputuloksensa aikaan saatiin toimiva, yksinkertainen ja helposti skaalautuva ratkaisu, joka palvelee myös tulevia integraatiotarpeita tuotekehitysyksikössä.

ASIASANAT:

Integraatio, Tietokantajärjestelmä

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme | Database Systems

June 2010 | 36

Minna-Kristiina Paakki

Petri Havia

Integration development project between two database systems

The purpose of this thesis is to describe an integration development project. The project's target was to design, develop and implement automated integration between two database systems used in company research and development units. The need of data transfer was pointed to sample dates that were maintained in both systems and synchronization was seen as critical from information visibility and quality point of view.

After the requirement gathering project was established as a joint effort between R&D and IT management and it was carried out mainly with internal resources. My responsibility area in this project included requirement management, concept development, testing and end user training. In these tasks we used process and data modeling tools that are used in company.

Integration of two systems with different platforms was challenging but rewarding effort and it was carried out in line with company standards and according architectural guidelines. As an end result a quite simple and scalable integration solution was implemented. This solution will serve also incoming integration needs in R&D

KEYWORDS:

Integration, Database system

Lyhenteet

PHJ	Portfoliohallintajärjestelmä. R&D Roadmap- ja portfoliohallinta työkalu
VHJ	Vaatimustenhallintajärjestelmä. Työkalua käytetään vaatimusten luontiin, seurantaan ja raportointiin
IT	Information Technologies. Vastaa globaalisti yrityksen tietoteknisten ratkaisujen saatavuudesta ja tuesta
R&D	Tuotekehitysyksikkö
ASW	Adaptation Software. SW kehitystiimi.
RF	Radio Frequency. RF tuotekehitystiimi
HW	Hardware. Yleisnimi laitteistolle, kuten matkapuhelimen koneisto
AIP	Application Integration Platform. Integraatiopalvelu.
API	Application Platform Interface. Sovelluksen itegraatorajapinta.
XML	eXtensive Markup Language. Tiedon siirrossa käytettävä dataformaatti.
OE	Operating Environment, jossa hallitaan yrityksen sisäisten käyttäjien ja systeemien tunnuksia
SSO	Single Sign On. Toiminnallisuus joka mahdollistaa kirjautumisen applikaatioihin ilman erillistä käyttäjätunnuksen ja salasanan syöttöä.
COP	Corporate Program management guidelines. COP tarjoaa suositukset ja ohjeet IT ratkaisuiden suunnitteluun ja käyttöönottoon sekä valmiit dokumenttipohjat
DSP	Domino Service Platform. Lotus Domino Server integraatioiden toteuttamisen.

SISÄLTÖ

LYHENTEET	4
1 JOHDANTO	7
1.1 Tausta	7
1.2 Tavoitteet	5
1.3 Työn rakenne	5
2 LÄHTÖKOHDAT	6
2.1 Toimintaympäristö ja tavat	7
2.2 Loppu- ja avainkäyttäjät	8
2.3 Paikalliset konseptiomistajat	8
2.4 Globaali konseptinomistaja	9
2.5 Tuki	10
2.6 Konfiguraatiotiimi	10
2.7 Asiakasorganisaatiot	11
2.8 Tietojärjestelmät	11
2.8.1 PHJ	11
2.8.2 VHJ ja AIP	13
3 PROJEKTIN ALUSTUS	14
3.1 Vaatimusten määrittely	14
3.2 Arkkitehtuurin valinta	15
3.3 Käyttötapausten jatkokehitys	17
4 PROJEKTIN KULKU	20
4.1 Projektin perustaminen, vaatimusten ja arkkitehtuurin hyväksyntä	20
4.2 Tiedonsiirto	22
4.3 Tiedon käyttö järjestelmissä	24
4.4 Testaus	26
4.5 Käyttöönotto	27
4.6 Ylläpitovaiheeseen siirtyminen ja projektin päättäminen	28
4.7 Projektin aikana opittua	29
5 LOPPULAUSE	30
LÄHTEET	31

LIITTEET	33
-----------------	----

KUVIOT

Kuvio 1 Tukimalli	7
Kuvio 2 Käyttötapaukset haastatteluista, (N.N.(1), 2009; N.N.(2), 2009)	15
Kuvio 3 Publish and Subscribe	17
Kuvio 4 Malliaikataulujen tietovirta PHJ → VHJ	19
Kuvio 5 Malliaikataulujen tietovirta VHJ → PHJ	19
Kuvio 6 Projektin aikataulu (N.N.(1), 2009)	21
Kuvio 7 Ympäristöjen viestinvälityslogiikka	23
Kuvio 8 Integraatio status määrittäminen	24
Kuvio 9 Linkitys PHJ:ssa	25
Kuvio 10 Linkityksen näkymä loppukäyttäjälle	25

TAULUKOT

Taulukko 1 Käyttötapausdokumenttien sisältö (COP, 2009)	18
Taulukko 2 Projektin vaiheet ja niiden sisältö	21

1 Johdanto

1.1 Tausta

Yritys, jolle opinnäytetyö tehtiin, on korkean teknologian yritys, jonka tuotekehitys ja myynti toimivat useissa eri maissa. Yritys tuottaa korkealaatuisia tuotteita laajan asiakaskunnan tarpeisiin, hyödyntäen ja tukien viimeisimmän teknologiakehityksen trendejä.

Vuoteen 2009 saakka yrityksessä oli oma yksikkö, joka kehitti keskitetysti komponentteja ja kokonaisuuksia kehitysprojektien käyttöön. Yksikköön oli koottu kaikki erityisosaaminen esimerkiksi kameroiden, näyttöjen ja äänentoiston kehityksestä. Yksikkö seurasi teknologiakehitystä maailmanlaajuisen verkoston avulla ja kehitti yhteistyössä alihankintaverkoston kanssa sovelluksia eri osa-alueille ja varmisti tällä tavalla kilpailukykyisen ja kattavan sisäisen tarjonnan. Tällä tavalla kehitysprojektien kehitystyö saatiin minimoitua ja ne saattoivat keskittyä valmiiden ratkaisujen integrointiin tuotteisiin ja tätä kautta kehitysprojektien aikataulut saatiin optimoitua minimiin. Haasteena suuren yksikön osalta oli tietojen ajantasainen hallinta ja jakaminen laajan tuotekehitysverkoston kanssa. Jotta tuotekehityksen aikataulujen ja kapasiteetin hallinta olisi ollut mahdollisimman tehokasta, tuli tiedonhallinnalla ratkaisujen tukea integroitua tiedonsiirtoa järjestelmien välillä. Haastetta tiedon vaihtamisessa eri järjestelmien välillä tuo myös erilaiset tekniset ratkaisut ja tiedon formaatti.

Tuotekehitysyksikkö hallinnoi itsenäisesti tuotekehityksessä käytettäviä IT-ratkaisuja, sekä tukea ja kehittää niitä yhdessä IT:n kanssa. Tuotekehitysyksikkö kehittää ratkaisunsa tuotekehityslähtöisesti ja varmistaa siten parhaan mahdollisen lopputuloksen ja käytettävyyden. Hallinto-yksikköön on keskitetty kaikki yhtiön yhteiset toiminnot, kuten tietotekniset palvelut,

henkilöstöhallinta ja lakitoiminnot. Hallinto-yksikköön kuuluva IT vastaa atk-laitteiston ympäristöistä, ylläpidosta ja eri käyttäjätasojen tuesta.

1.2 Tavoitteet

Tämä työ on toiminnallinen opinnäytetyö ja sen aineena olleen projektin tavoitteena oli selvittää vaatimukset, mallintaa toteutuskonsepti sekä rakentaa ja testata toteutus. Lisäksi projektin tavoitteisiin kuului tuottaa tarvittava dokumentaatio ylläpito- ja käyttäjäorganisaatiolle sekä kouluttaa loppukäyttäjät. Oma osuuteni sisälsi vaatimusten keräämisen, tarkentamisen ja toteutuskonseptin rakentamisen.

Tärkeää oli ymmärtää ja kartoittaa vaatimusten takana olevat prosessit ja samalla tarkastella ja kehittää niitä tehokkaammiksi aina kuin mahdollista. Järjestelmäkehityksen lähtökohtana tulisi aina olla olemassa oleva toimintatapa tai prosessi, jota tuetaan työkaluratkaisulla, mutta monissa tapauksissa kehitys tehdään rinnakkain kummallakin alueella. Tällaisissa tapauksissa iteratiivinen kehitys on usein tehokkain tapa saada haluttu lopputulos aikaiseksi.

Tämän työn tavoitteena oli suunnitella ja toteuttaa integraatio kahden tuotekehitysyksikössä käytössä olevan tietokantajärjestelmän välillä ja parantaa tätä kautta suunnittelun läpinäkyvyyttä ja parantaa aikataulullista performanssia.

1.3 Työn rakenne

Tämä työ esittelee yleisiä toimintatapoja ja tukiorganisaatiota työn tilaajana olleessa yrityksessä ja aiheena olleen projektin etenemistä eri vaiheissa omaa vastuualuettani silmällä pitäen. Työssä esitellään keskeisimmät konseptit IT-kehitysprojekteissa ja käyttöympäristöissä sekä kyseisen projektin eteneminen ja lopputulos.

2 Lähtökohdat

Tässä luvussa kuvataan yrityksen organisaatio projektin aikana, IT-palvelujen tukimallista ja siihen kuuluvien toimijoiden vastualueet.

Tuotekehitys toimi aikana, jolloin opinnäytetyön aiheena oleva projekti toteutettiin, niin kutsutulla Platform-mallilla. Tässä mallissa tuotteen eri osa-alueet suunniteltiin ja toteutettiin keskitetysti nimetyissä tuotekehityksen yksiköissä ja tuotteistettiin kokonaisuuksiksi kehitysprojekteissa. Tämän toimintatavan periaatteena oli keskittää erityisosaaminen suurempiin yksiköihin, joissa itseohjautuvasti kohdistettiin resurssit tekemään niitä hyödykkeitä, joilla katsottiin olevan suurin potentiaali tuoda paras mahdollinen lopputulos. Myös huippuosaamisen keskittäminen nähtiin tämän toimintamallin vahvana puolena. Kun kaikki saman alueen ihmiset toimivat yhdessä saadaan tietoa jaettua tehokkaasti ja kaikilla on hyvä näkemys meneillään oleviin kehitysaktiviteetteihin. Esimerkiksi kameroiden, akkujen ja muistikomponenttien kehityksessä keskityttiin uusien teknologioiden tuomiin mahdollisuuksiin pienentää tarvittavaa tilaa lopputuotteessa tai kasvattaa loppukäyttäjällä käytössään olevaa kapasiteettia tallentaa kuvaa tai ääntä parhaassa mahdollisessa muodossa sekä pidentää päätelaitteen toiminta-aikaa.

Tämän toimintamallin haasteena oli synkronoida eri toimitusten aikataulut siten, että kokonaisuus saatiin valmiiksi yhtä aikaa. Tätä kokonaisuuden aikataulua ohjasi ja valvoi tuotehanke ja sen johtaja. Johtuen aliprojektien erilaisista kypsyysraportointimalleista ja kirjavista dokumentointi- ja järjestelmäkäytännöistä, seurantaan tarvittiin paljon manuaalista työtä ja tarkistuksia puhelimitse ja muilla kommunikointimenetelmillä. Voikin sanoa, että tuotehankkeen johtajan piti tietää ja tuntea kaikki osa-alueet erittäin tarkisti voidakseen seurata kehitystyön etenemistä luotettavasti. Tieto on yksi olennainen osa menestyksestä tuotekehitystä, liittyi se sitten itse tuotteeseen tai sen tekemiseen ja aikataulutukseen.

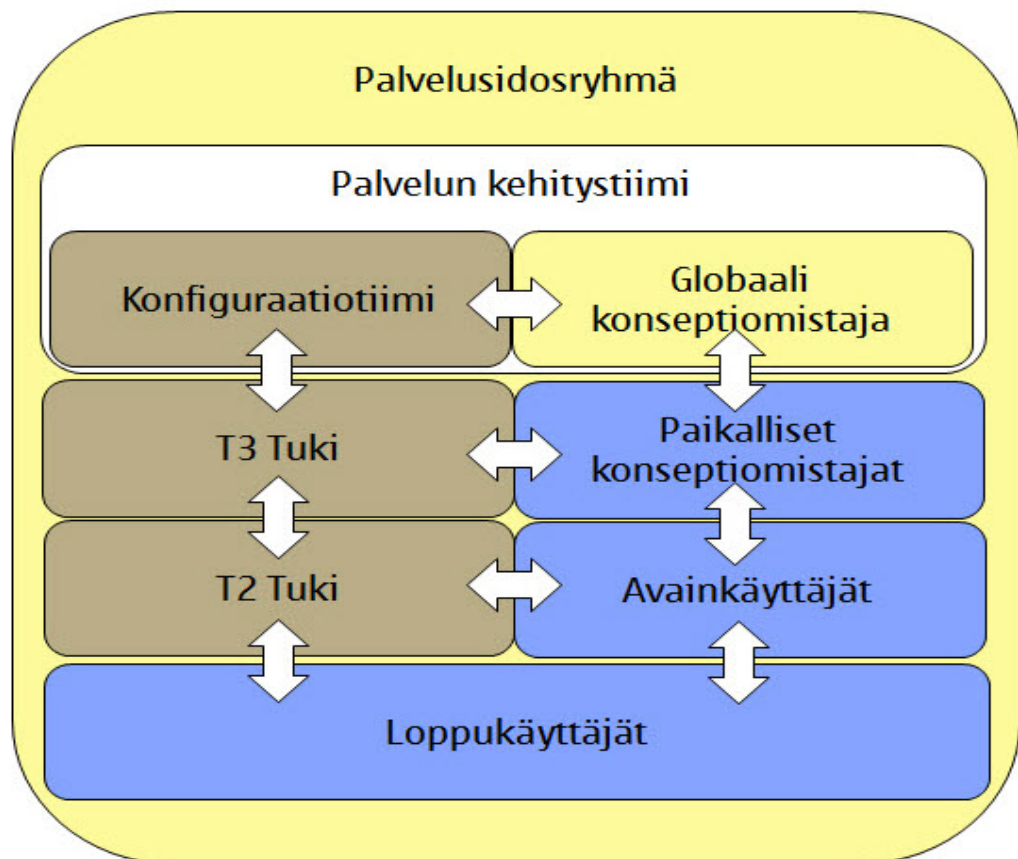
Kaksi käytössä ollutta keskeistä tietojärjestelmää tuotekehityksessä olivat VHJ (Vaatimustenhallintajärjestelmä), jossa hallinnoitiin kehitysvaatimuksia ja niiden

toteutukseen liittyviä tehtäviä, ja PHJ jossa hallinnoitiin teknologiaelementtien teknistä- ja aikataulullista dataa.

Näiden kahden järjestelmän välillä nähtiin selkeä tarve päivittää tietoja, jotta se olisi ajantasaista molemmissa järjestelmissä ilman kahteen kertaan syöttämistä. Manuaalinen ylläpito ja päivitysten unohtaminen tai laiminlyönti saattaisi aiheuttaa vakavia viivästymisiä tuotehankkeille.

2.1 Toimintaympäristö ja tavat

Yrityksessä tietohallintaratkaisut seuraavat pääsääntöisesti niin kutsuttua globaalia tukimallia, joka koostuu kolmitasoisesta tukiorganisaatiosta (Kuva 1).



Kuvio 1 Tukimalli

Tällä mallilla mahdollistetaan suurien käyttäjämäärien hallinta, ohjaaminen, koulutus ja informaation tehokas jako. Osallistuvat tahot, toimintaperiaate ja vastuut eri tasoilla on selitetty seuraavassa kappaleessa.

2.2 Loppu- ja avainkäyttäjät

Loppukäyttäjäksi lasketaan henkilöt, jotka käyttävät palvelua joko tiedon tallentamiseen tai lukemiseen. Tyypillistä loppukäyttäjille on, että he eivät kouluta tai kehitä palvelua, vaan keskittyvät sen sisältöön. He voivat kuitenkin ehdottaa muutoksia tai kehityskohteita järjestelmässä lähimmälle avainkäyttäjälle.

Avainkäyttäjiä ovat edistyneemmät käyttäjät, joilla on kokemusta oman osaamisalueensa toiminnallisuuksista ja prosesseista. He toimivat omissa yksiköissään ensisijaisina kontakteina loppukäyttäjille ja osaavat antaa apua niin järjestelmään, kuin sen sisältöön liittyvissä kysymyksissä. Avainkäyttäjät vastaanottavat kehitystarpeita loppukäyttäjiltä ja analysoivat tarpeen omasta näkökulmastaan. Joissain tapauksissa vaatimukset ovat toteutettavissa jo olemassa olevilla toiminnallisuuksilla ja silloin avainkäyttäjät opastavat loppukäyttäjää ratkaisemaan ongelman itsenäisesti. Jos ratkaisua ei voida toteuttaa olemassa olevalla toiminnallisuudella ja avainkäyttäjä katsoo vaatimuksen oleelliseksi, avaa hän siitä vaatimustenhallintajärjestelmään pyynnön ja liittyy siihen vaadittavat tarkennukset ja dokumentaation tarpeesta.

Avainkäyttäjät kouluttavat oman loppukäyttäjäjoukkonsa uusien toiminnallisuuksien ja prosessimuutosten osalta ja valvovat tiedon eheyttä omalla alueellaan.

2.3 Paikalliset konseptiomistajat

Paikallinen konseptiomistaja voi olla joko yksikkö-, maanosa- tai toimipaikkakohtainen. He toimivat oman yksikkönsä ja/tai prosessialueensa asiantuntijoina ja toimivat asiakasorganisaation edustajina palvelukehityksessä.

Tässä roolissa he vastaanottavat vaatimuksia avainkäyttäjiltä, analysoivat ja tarvittaessa tarkentavat niitä ja joko hyväksyvät tai hylkäävät ehdotukset. Hyväksytyt ehdotukset nostetaan vaatimustenhallinnan seuraavalle tasolle ja hylätyt ehdotukset kommunikoidaan avainkäyttäjille.

Konseptiomistajat omaavat oman alueensa asiantuntijuuden käytössä olevien työkalujen ja prosessien osalta, vastaavat oman avainkäyttäjäorganisaationsa kouluttamisesta ja toimivat aktiivisesti viestinviejinä sekä omaan että palveluorganisaatioon. Tällä toimintatavalla varmistetaan paras mahdollinen ymmärrys päivittäisen työn tukemiseen ja sen kehittämiseen.

Loppukäyttäjät, avainkäyttäjät ja paikalliset konseptiomistajat kuuluvat organisaation osiin, joiden ydintoimintaa myyntihyödykkeiden kehittäminen, valmistus tai markkinointi. Tätä kautta heitä voi luonnehtia tietoteknisen palvelun sisäisiksi asiakkaiksi.

2.4 Globaali konseptinomistaja

Globaali konseptinomistaja vastaanottaa vaatimuksia paikallisilta konseptinomistajilta ja tarvittaessa täydentää ja jatkokehittää niitä, sekä hyväksyy tai hylkää ehdotukset. Hän kehittää toteutuksen konseptin hyväksytyille ehdotuksille ja dokumentoi sen järjestelmävaatimuksen muotoon. Tämän lisäksi hän tuottaa tarvittavat prosessi- ja käyttötapauskuvat sekä katselmoi ne vaatimusten omistajien kanssa ennen toteutusta. Hän vastaa järjestelmästä kokonaisuutena yli asiakasorganisaatioiden ja toimii yhteistyössä IT organisaation kanssa sekä suunnittelee ja johtaa kehitystyötä palvelutiimin sisällä. Globaalin konseptinomistajan rooliin kuuluu kommunikointi ja paikallisten konseptinomistajien kouluttaminen järjestelmä- ja prosessi muutosten osalta. Organisaatiossa globaali konseptiomistaja kuuluu tyypillisesti riippumattomaan organisaation osaan, jotta voi puolueettomasti ja kokonaisuuden kannalta mahdollisimman tehokkaasti tuottaa toteutusehdotuksia, jotka palvelevat kokonaisuutta parhaiten.

2.5 Tuki

T2 tiimi luo uudet käyttäjätunnukset ja ylläpitää oikeusmäärittämiä tarpeen mukaan. Nämä tukihenkilöt toimivat loppukäyttäjien ja avainkäyttäjien teknisenä tukena järjestelmän käyttöön liittyvissä ongelma- ja vikatilanteissa. Koko yrityksen kattavan palvelupisteen kautta tulevat vikailmoitukset ja palvelupyynnöt ohjataan raportointijärjestelmään, jota kautta T2 tuki saa ilmoituksen palvelupyynnöistä.

T3, eli edistynyt tekninen tuki, on tiimi, joka valvoo ja ylläpitää tietokantaa ja sen palveluita. He toimivat kontaktina serverien ylläpito-organisaatioon ja muihin palveluihin tarpeen mukaan. Tällaisia ovat mm. toiset palvelut, joiden kanssa on olemassa olevaa tiedonsiirtoa tai yhteistoimintaa.

T3 suorittaa tarvittaessa tietokannan sisältöön liittyviä tiedonsiirtoja, kyselyitä ja tuottaa tietokantaan tai sen sisältöön liittyviä raportteja. Lisäksi he koordinoivat ja vastaavat kehitysversioiden testaamisesta eri ympäristöissä ja niiden käyttöönotosta tuotantoympäristöön.

2.6 Konfiguraatiotiimi

Konfiguraatiotiimi toteuttaa ja testaa uusien konseptien käyttöönottoa tietokannassa tai sen konfiguraatiokerroksessa. Tiimi toimii globaalien konseptiomistaja ohjauksessa, mutta kehittää myös itsenäisesti toteutus- ja parannusehdotuksia tietokannan rakenteeseen, konfiguraatioon ja eheyteen. Konfiguraatiotiimin jäsenet ovat oman alueensa asiantuntijoita järjestelmän toiminnallisuuksien osalta, mutta kaikilla on vähintään tyydyttävä ymmärrys kokonaisuudesta. Tiimin jäsenten osaamista kehittää ja ohjaa tyypillisesti tiimin kokenein jäsen.

T2- ja T3 tuki sekä konfiguraatiotiimi ovat osa IT organisaatiota ja siten eivät suoraan asiakasorganisaation ohjauksessa.

2.7 Asiakasorganisaatiot

Alkuperäisten vaatimusten takana oli joukko yksittäisiä henkilöitä ja tarkemmassa läpikäynnissä tunnistettiin mukana olevan kolme tahoja:

- Adaptation Software (ASW), joka tuottaa ohjelmistoratkaisut tuotteen eri osien yhteensovittamiseksi. Tarvitsee näkyvyyden eri osien tarkkaan mallikappaleaikatauluun, jotta se voi tehokkaasti suunnitella ja toteuttaa ohjelmistokehitystään. ASW ylläpitää toteutusaikatauluaan VHJ tietojärjestelmässä.
- Display, joka keskitetysti kehittää näyttöjä, luo näkyvyyttä olemassa olevaan ja tulevaan tarjontaan ja aikatauluihin PHJ:ssa
- RF Team, joka tuottaa RF-ratkaisuja. Tiimi pitää aikataulutietoa yllä VHJ:ssä ja teknologian sisältöä PHJ:ssä.

2.8 Tietojärjestelmät

Tämä luku kuvaa korkealla tasolla projektin kohteena olleiden tietojärjestelmien ja palveluiden ominaisuuksia ja käyttötarkoituksia.

2.8.1 PHJ

PHJ on kaupallisen ohjelmistotoimittajan tuoteperheeseen kuuluva, tuote- ja portfoliohallintaan suunnattu työkalu. Tietokantana Oracle 10, Linux servereillä. Järjestelmässä on API standardi 2.0 rajapinta, jonka päälle on itse rakennettu integraation asiakasohjelma (eng. Client).

Yritys käyttää kyseistä työkalua tuoteskenaarioiden suunnitteluun, portfoliohallintaan ja aikataulusuunnitteluun. Tietosisällön fokus on suunnittelun ja päätöksenteon kannalta kriittisissä tietoalkioissa ja aikamielessä tästä päivästä eteenpäin. Suurin käyttäjäorganisaatio on tuotekehitys sidosryhmineen, mutta myös muissa osissa yritystä tätä tietoa tarvitsevat henkilöt voivat saada käyttöoikeuden järjestelmään.

Sisältö on jaoteltu teknologia perusteisesti omiin työtiloihin ja loogisiin kokonaisuuksiin. Tarkoituksena tällä on rajata sisältö pienempiin osiin ja mahdollistaa tarvittaessa myös rajoitetumpi näkyvyys käyttäjille.

Käyttäjähallinta tapahtuu tietokantatasolla ja käyttöoikeushallinta työtila tasolla. Käyttäjätunnuksena käytetään yrityksen OE (Operating Environment) käyttäjätunnusta ja kirjautuminen tapahtuu SSO:n kautta, jolloin erillistä applikaatitason kirjautumista ei tarvita, vaan järjestelmä tunnistaa käyttäjän intranet-istunnon kautta. Aktivoidulla käyttäjätunnuksella pääsee kirjautumaan sisään palveluun, mutta se ei yksistään tuo näkyvyyttä informaatioon.

Yksittäinen ilmentymä PHJ:ssä tunnetaan elementtinä, joka pohjautuu moduuliin. Moduulitasolla määritellään käytössä olevat attribuutit ja yksittäisen ilmentymän käyttäytyminen. Tämä on hyvin lähellä perinteistä relaatiotietokantaa, jossa tauluun määritetään ilmentymien sisältöä ja käyttäytymistä ohjaavat määrittymät, kuten attribuutit. Suurin ero näiden välillä on, että PHJ:n taulut ovat kiinteät, jolloin moduulit ja elementit muokataan niihin dynaamisesti, ei tauluja luomalla tai manipuloimalla kantatasolla.

Käyttäjien oikeudet järjestelmän tietoihin, niiden käyttöön ja manipulointiin, hallitaan työtilatasolla. Käyttäjätili liitetään ryhmään, jonka kautta tämä saa käyttöönsä näkymiä. Näkymät ovat ennalta määritettyjä ja niihin on määritelty, mitkä taulut ja attribuutit ovat näkyvissä ja ovatko ne editoitavissa vai lukutilassa. Näkymiä voidaan liittää myös suoraan käyttäjätiliin, mutta suurin osa niistä periytyy ryhmien kautta, joihin käyttäjä kuuluu.

Integraatorajapinta on standardi API 2.0, joka on joustava ja sen käyttäytyminen määritellään pitkälti integraatioasiakasohjelmalla (Engl. Integration Clint). Tällä ohjelmalla voidaan määrittää mm. siirtojen taajuus, sisältö ja kohde.

Järjestelmän käyttö tapahtuu internet-selaimella ja paikallisia, tietokonekohtaisia asennuksia ohjelmistolle ei tarvita. PHJ:n käyttäjämäärä yrityksessä on noin 5000.

2.8.2 VHJ ja AIP

VHJ on Lotus Notes pohjainen järjestelmä tuotekehitykseen. Pohjana on Lotus Domino ratkaisu.

Käyttö keskittyy vaatimusten hallintaan, seurantaan ja raportointiin. Vaatimustasolla seurataan kehittymistä ja aikatauluja eri osakokonaisuuksien tasolla ja tietoa tuotetaan ja jaetaan eri yksikköjen välillä.

Tiedon siirto tapahtuu DSP:n kautta sisään ja ulos kulkevien viestijonojen muodossa. Perusasetuksena jonot ladataan 10 minuutin välein. Muuttunut ja integroitavaksi merkitty tieto (delta) siirretään DSP serverille, josta se lähetetään tai noudetaan toiseen palveluun. Tätä samaa toiminnallisuutta käytettiin myös integraatiossa PHJ:n kanssa. Järjestelmä on ollut käytössä useita vuosi yrityksen tuotekehityksessä ja sillä on haasteita vanhahtavan teknisen toteutuksensa vuoksi.

AIP on IT:n tarjoama alusta järjestelmien integroimiseen. Se toimii viestin välittäjänä ja varastointipaikkana. Myös viestin manipulointi on mahdollista, jos se on tarpeellista lähettävän ja vastaanottavan pään yhteensovittamiseksi. Perinteisesti kuitenkin viestit on pyritty tuottamaan sellaisena, ettei sitä tarvitse muuttaa lähetyksen aikana.

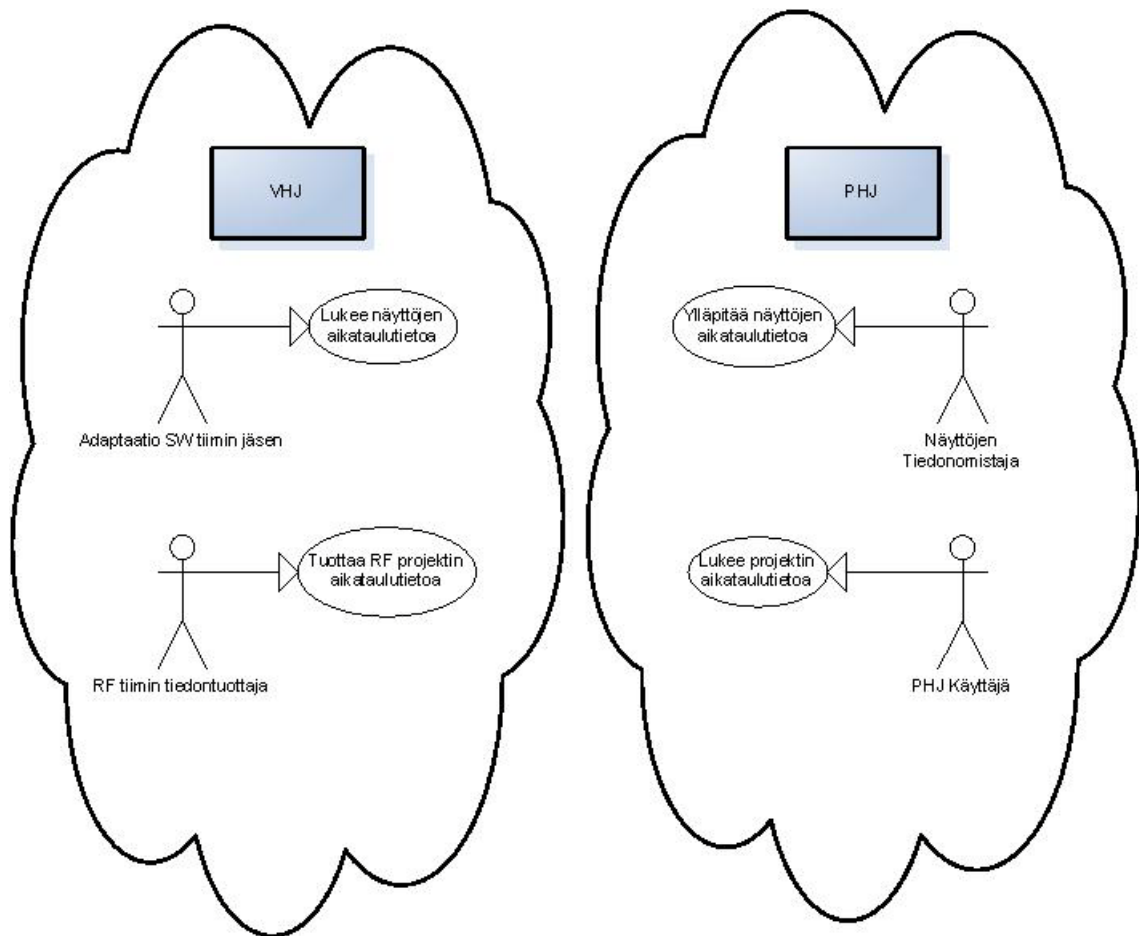
3 Projektin alustus

Tässä osassa kuvataan projektin aloittamiseen tähtäävän suunnittelun osuus. Tarkoitus tälle oli saada riittävän selvä käsitys vaatimuksista, niiden tarpeellisuudesta ja toteutettavuudesta. Lisäksi valmisteltiin hahmotelma mahdollisesta teknisestä arkkitehtuurista ja toteutuksesta

3.1 Vaatimusten määrittely

Koska tarpeiden vaatimusmäärittelyä oli jo tehty kummankin mukana olevan työkalun tukiorganisaatiossa (N.N.(2), 2009; N.N.(3), 2009) oli ensimmäinen vaihe tunnistaa osalliset ja listata tiedossa olevat tarpeet ja tarkistaa mahdolliset päällekkäisyydet ja puutteet. Kummankin palvelun globaalit konseptinomistajat keräsivät saatavilla olevat vaatimukset ja kirjasivat ne yhteiseen vaatimusdokumenttiin. Toteutuksen ensimmäiset ajatukset kirjattiin konseptiehdotukseen ja kommunikaatiomateriaali luotiin keskustelun tueksi. Keskustelut osapuolien kanssa aloitettiin ensimmäisen hahmotelman tukemana ja sitä tarkennettiin yksityiskohtaisemmaksi. Tavoitteiden selkiytyessä luotiin tähän pohjautuen projektiehdotus.

Vaatimuksista tunnistettiin neljä keskeistä käyttötapausta.



Kuvio 2 Käyttötapaukset haastatteluista, (N.N.(1), 2009; N.N.(2), 2009)

Projekti sai aloitus- ja resursointiluvan heti ensimmäisessä katselmoinnissa, jossa samalla päätettiin nimittää projektipäällikkö ja ohjausryhmä. Projektiryhmän edustajat valittiin osallisista käyttäjäryhmistä ja palveluista. Tässä vaiheessa jatkoi poikkeuksellisesti kaksi rinnakkaista polkua. Toinen aloitti resursoinnin suunnittelun projektipäällikön haulla ja toinen aloitti ajan säästämiseksi arkkitehtuurin määrittämisen ja käyttötapausten tarkennuksen.

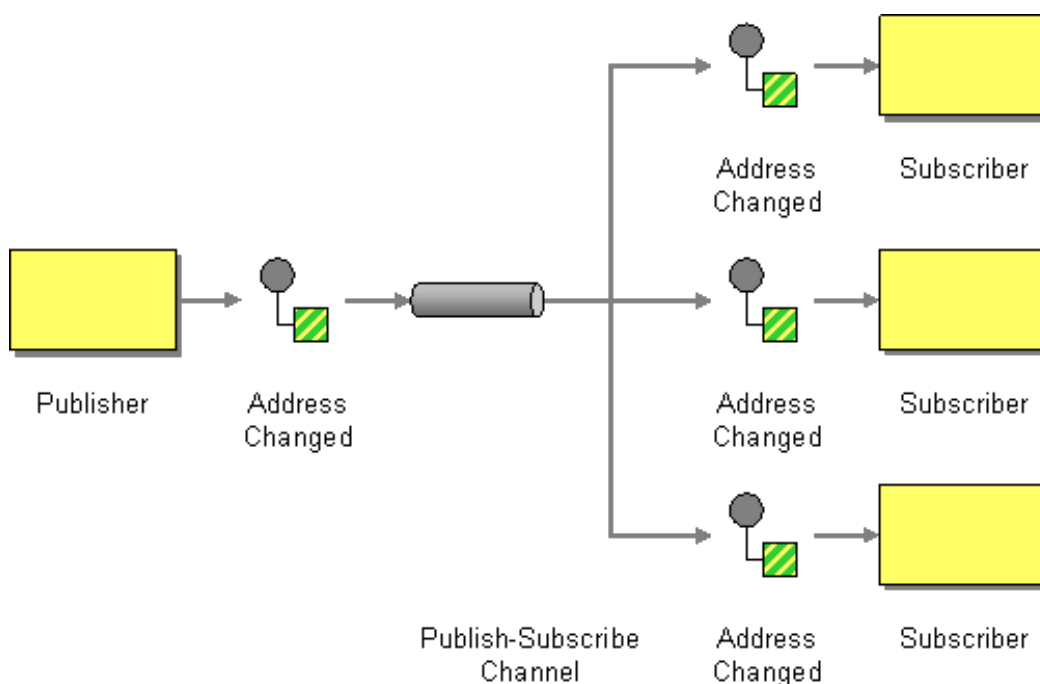
3.2 Arkkitehtuurin valinta

Yrityksen tietoteknisten ratkaisujen ohjeistusta antaa arkkitehtuuritiimi, joka tuottaa suosituksia informaatorakenteiden ja teknisten ratkaisujen osalta. Palveluiden konseptiomistajat keskustelivat arkkitehdin kanssa eri ratkaisumalleista.

Keskustelujen perusteella integraatio päätettiin toteuttaa julkaise ja tilaa periaatteella (eng. Publish and Subscribe) (Markus Valo, 2009), jossa tiedon omistava järjestelmä julkaisee tiedot sovittuun paikkaan ja vastaanottava tieto järjestelmä noutaa sen. Toimintatapaa voi kuvata yksinkertaisesti postin välittämisenä siten, että kirjeen kirjoittaja toimittaa kirjeensä postilaatikkoon, josta sen vastaanottaja(t) käyvät kopioimassa sen joko kokonaan tai osin, tarpeensa mukaan. Kirjeen toimittaja on antanut luvan tai avaimen postilaatikon käyttöön vain tietyille tahoille ja sivullisilla ei ole mahdollista noutaa sieltä tietoja. Välittävä palvelu on formaatti-vapaa, joten välitettävä tieto voi olla monessa muodossa. Ratkaisuna tämä tapa vähentää myös saman tiedon siirtämistä moneen kertaan eri järjestelmiin.

IT tarjoaa tähän valmiin integraatiopalvelun, AIP (Application Integration Platform). AIP mahdollistaa myös informaation manipuloinnin siirron yhteydessä. Tämä on käytännöllinen ratkaisu, jos integroitavat tietokannat ovat rajoittuneita tiedon manipuloinnin tai formaatin suhteen. Tässä projektissa todettiin kuitenkin lähetettävien viestimuiden olevan sellaisia, että manipulaatiota ei tarvittu.

Toisaalta viestin manipulointi aiheuttaa myös lisää testausta ja riskejä ylläpitovaiheessa, jos sisältö tai viestimuoto muuttuu.



Kuvio 3 Publish and Subscribe

3.3 Käyttötapausten jatkokehitys

Neljän aluksi tunnistetun käyttötapausten tarkemman määrittämisen yhteydessä valittua arkkitehtuuria vasten, kävi nopeasti selville, että niiden toteuttaminen aiheuttaa muutoksia nykyisiin järjestelmien prosesseihin ja käyttötapauksiin. Lisäksi mukaan otetut tekniset henkilöt antoivat ohjeistusta toteutuksen rajoituksista ja mahdollisuuksista. Tämän jälkeen kirjoitettiin uudet käyttötapausten informaation tuottamisen ja siirtämisen näkökulmasta, käyttäjälähtöisyyden sijaan.

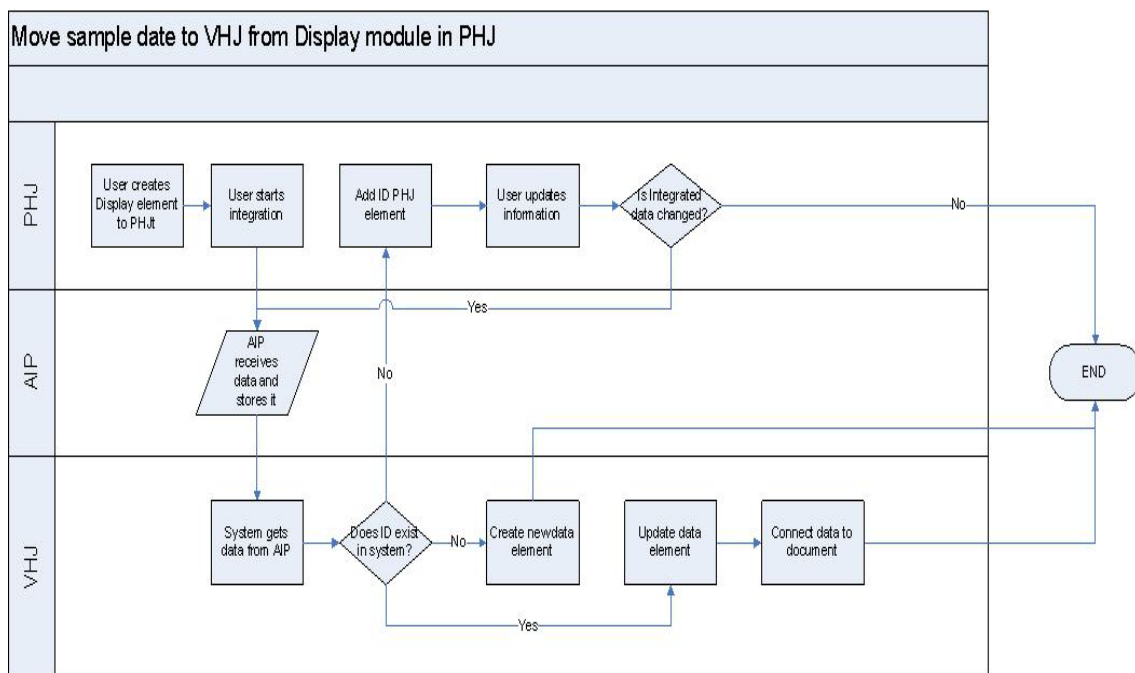
Dokumentaatio toteutettiin yrityksen sisäisen standardin (COP, 2009) mukaisella mallilla. Jokainen tietovirta kuvattiin askel askeleelta siten, että se kuvasi käyttäjän tekemät transaktiot sekä toivotun järjestelmävasteen. Tietovirtana tunnistettiin kaksi käyttötapausta. Käyttötapaustodokumenteissa kuvataan taulukossa 1 listatut asiat.

Taulukko 1 Käyttötapausdokumenttien sisältö (COP, 2009)

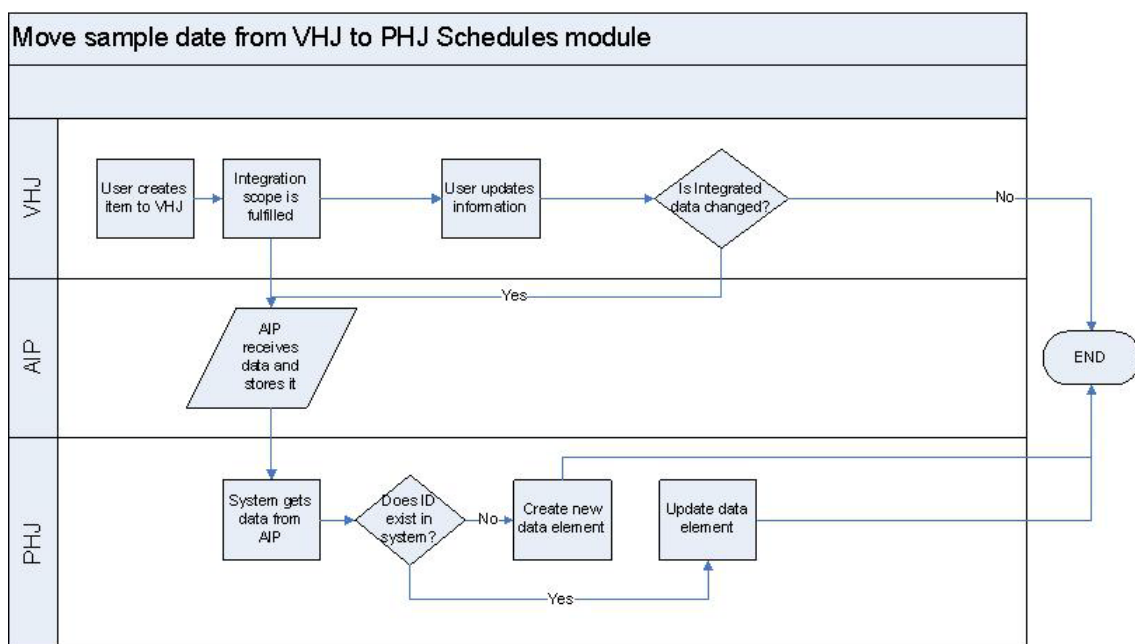
Yleiskuvaus käyttötapauksesta, joka on lyhyt kuvaus käyttötapausten sisällöstä ja kohteesta
Laajuuden määrittäminen, jossa rajataan mukaan kuuluvat asiat.
Toimijoiden roolikuvaukset kertovat siitä, millaisella roolilla toimivia henkilöitä tai järjestelmän automaatioita kuuluu käyttötapausten piiriin
Suunnitteluoletukset. Mikäli oletetaan tiettyjen raja-arvojen tai toimintojen olevan olemassa, ne kuvataan dokumenttiin tai niihin viitataan
Tapahtumien kulku askeleittain. Tämä on kuvauksen tärkein vaihe. Tähän kirjoitetaan askel askeleelta käyttäjän tekemät valinnat ja odotukset siitä, miten järjestelmä niihin vastaa. Viittaukset vaihtoehtoihin tai poikkeuksellisiin toimintoihin kirjataan tähän osaan.
Vaihtoehtoiset toimintatavat. Jos käyttäjällä on useampi mahdollinen lopputulos tekemälleen valinnalle, kirjataan ne tarkemmin tähän osaan. Viittaukset tehdään tapahtumien kulku kohtaan.
Säännöstö. Jos on olemassa säännöstöjä, joita käyttäjän ja järjestelmän pitää noudattaa, kuvataan se tähän osioon. Esim. Syötetty informaatio ei saa sisältää skandinaavisia kirjainmerkkejä.
Poikkeukset. Tähän kirjoitetaan poikkeavat ja odottamattomat tilanteet ja niiden käsittely. Esim. Jos järjestelmä ei pysty tallentamaan tietoa tuntemattomasta syystä, palauttaa se käyttäjälle virheilmoituksen.
Liitteet. Mahdolliset asiaan liittyvät materiaalit, kun alkuperäiset vaatimukset, jos ne ovat tapauksen kannalta oleellisia.

Esimerkki käyttötapausten kuvauksesta liitteessä 1.

Käyttötapausten prosessit kuvattiin myös kaavioina kommunikoinnin helpottamiseksi. Prosessit muistuttivat ideologialtaan läheisesti toisiaan. Suurin ero oli PHJ VHJ siirron manuaalinen käynnistäminen, kun toiseen suuntaan tehtävä siirto tehtiin perustuen ennaltamäärätyihin kriteereihin.



Kuvio 4 Malliaikataulujen tietovirta PHJ → VHJ



Kuvio 5 Malliaikataulujen tietovirta VHJ → PHJ

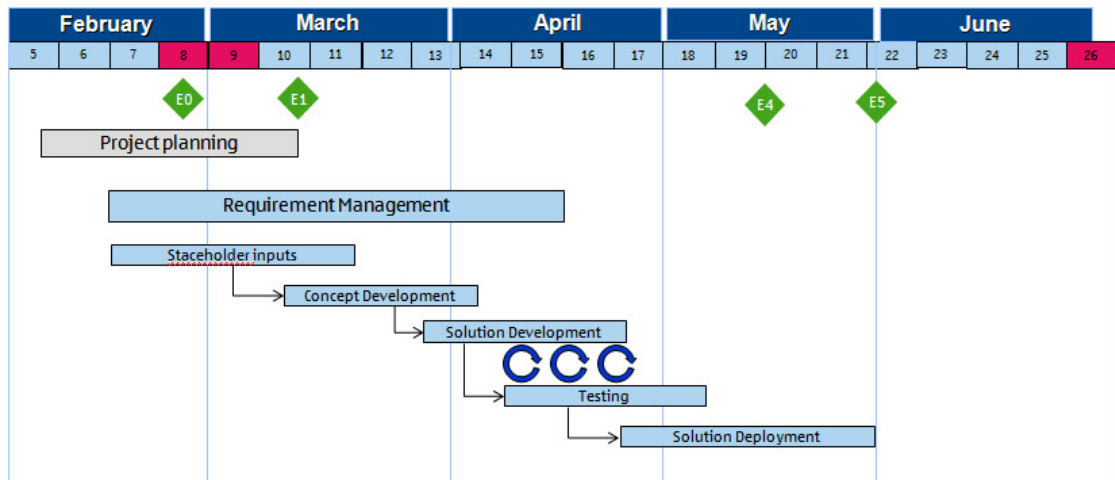
4 Projektin kulku

Tämä luku keskittyy kuvaamaan projektia ja toteutuksen tekemiseen liittyneitä asioita ja vaiheita. Kuten edellisessä kappaleessa mainittiin, valmistelutyö jatkui yrityksen projektikäytäntöjä etupainotteisemmin. Selostus on tehty PHJ-painotteisesti, koska en itse osallistunut VHJ kehitykseen, sen kuuluessa toiselle konseptiomistajalle.

4.1 Projektin perustaminen, vaatimusten ja arkkitehtuurin hyväksyntä

Toteutusvaiheen alkuun mennessä projektin organisaatio oli hahmoteltu ja projektipäällikkö aloitti varsinaisen ohjauksen. Henkilö projektipäälliköksi valittiin ulkopuolisesta palveluyrityksestä, koska projektin sisältö ei näyttänyt vaativan syvällistä teknistä osaamista kyseisessä roolissa. Projektiin kuului projektipäällikön lisäksi kummankin järjestelmän globaalit konseptinomistajat ja integraatiokehittäjät, AIP-asiantuntija ja neljä asiakasryhmän edustajaa. Projektin etenemisen seuranta päätettiin toteuttaa viikkosyklillä siten, että joka viikolle varattiin kahden tunnin seurantapalaveri, jossa käytiin läpi toteutuneet tehtävät ja suunniteltiin seuraavan viikon tehtävälistä. Projektipäällikkö vastasi tehtävien toteutuman seurannasta ja jokaiselle jäsenelle sovittiin raportointivastuu omasta alueestaan. Projektin aloittamisesta, valmistumisesta ja tulosten hyväksynnästä valittiin vastaamaan ohjausryhmä, joka koostui projektipäälliköstä ja asiakasryhmän edustajat. Aloitus päätettiin ensimmäisessä ohjausryhmän kokouksessa olemassa olevin määrityksin. Lopetuskriteeriksi sovittiin toimiva ratkaisu, joka toteuttaa olemassa olevat vaatimukset. Projektin toteutusaikatauluksi sovittiin kolme kuukautta. Karkea aikataulu projektille on kuvattu kuviossa 6.

Project Schedule



Kuvio 6 Projektin aikataulu (N.N.(1), 2009)

Koska projektin tuotos näytti edelleen suoraviivaiselta ja nopeaksi toteuttaa, päätettiin sen vaiheistus yksinkertaistaa yrityksen IT-projektien kuusivaiheisesta etenemismallista kolmeen vaiheeseen. Tällä päätöksellä oli tarkoitus yksinkertaistaa projektin päätöksentekoa. Vaiheiden sisällöksi sovittiin taulukossa 2 kuvatut asiat.

Taulukko 2 Projektin vaiheet ja niiden sisältö

E0-E1 Spesifiointi ja arkkitehtuuri. Tässä vaiheessa viimeistellään vaatimukset ja käyttötapaukset sekä valitaan toteutusarkkitehtuurin mukaiset palvelut. Tämä vaihe oli käytännössä jo valmistunut, koska vaatimusten osalta viimeistely oli valmis ja arkkitehtuurin oli valittu
E1-E4 Järjestelmäsuunnittelu, toteutus ja testaus. Tässä vaiheessa tehdään käytännön koodaus ja kehitystyö kaikkiin ympäristöihin. Testataan viestien kulkeminen järjestelmien välillä ja todennetaan viestin sisällön ja formaatin toimivuus.
E4-E5 Implementaatio ja koulutus. Uuden toiminnallisuuden käyttöönotto ja koulutus loppukäyttäjille.

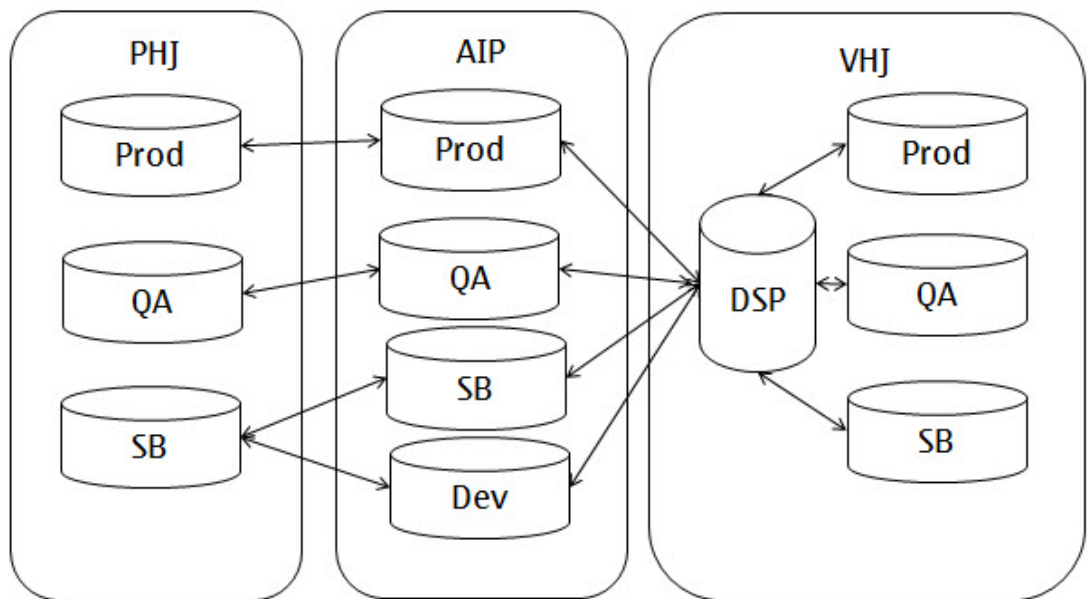
Kuten aloitusvaiheessa päätettiin, vaatimusten kehittäminen tapahtui normaalin vaatimustenhallinnan prosessien mukaisesti aina siihen saakka, kun projektille

nimettiin vastuullinen projektipäällikkö, joka otti vastuulleen kehitystyön ohjauksen ja resursoinnin. Tässä vaiheessa olemassa olevat kuvaukset olivat jo sellaisella tasolla, että ohjausryhmä hyväksyi ne ja projektin ensimmäinen vaihe aloitettiin.

4.2 Tiedonsiirto.

Suunnitelmien pohjalta aloitettiin kehitystyöt alimmissa ympäristöissä. Yrityksen tietokantapalvelut tukeutuvat vähintään kolmiportaiseen ympäristömalliin: Hiekkalaatikko (Engl. Sandbox (SB)), Laadunvarmistus (Engl. Quality Assurance(QA)) ja Tuotanto (Engl, Production (Prod)). Periaatteena on, että kaikki tehtävät muutokset kuljetetaan aina alhaalta ylöspäin, varmistaen näin toteutuksen toimivuus ja tehokkuus. Ensimmäisten työpajojen yhteydessä huomattiin, että mukana oleva järjestelmäjoukko ei ollut täysin yhtenäinen, johtuen integraatioalustan nelitasoisesta ympäristömallista. Tästä johtuen jouduttiin tekemään erillinen suunnitelma siitä, mitkä ympäristöt keskustelivat keskenään eri vaiheissa.

Lopputuloksena saatiin kuvaus siirtojen logiikasta ja sopimus siitä, että AIP käyttää hiekkalaatikkotason tietoja kahdessa ensimmäisessä testikierroksessa. PHJ ja VHJ kohdalla tämä tarkoitti vain kahta peräkkäistä siirtoa eri palvelinosoitteen kautta. VHJ käyttää omaa, erillistä integraatioserveriä, jonka kautta se toimittaa ja vastaan ottaa kaikkien ympäristöjen tiedon siirron. Viestien sisältöön kirjoitetaan vastaanottavan ympäristön palvelinosoite.



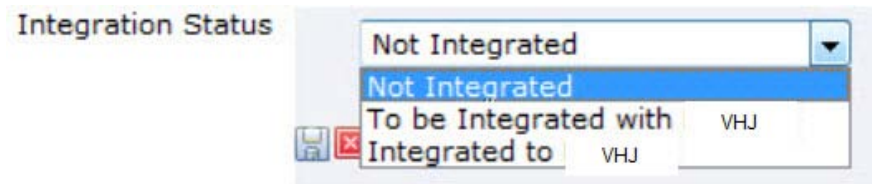
Kuvio 7 Ympäristöjen viestinvälityslogiikka

Työt aloitettiin kaikissa ympäristöissä varmistamalla viestiformaatin (XML) siirtyminen rajapintojen läpi ehjänä. Alussa viestin sisältö ei vielä vastannut tuotannollista informaatiota. Viesti lähetettiin manuaalisesti, ilman automaattisia laukaisintoimintoja, koska tässä vaiheessa riitti lähetys- ja vastaanottokyvyn toteaminen. Tuloksena voitiin todeta kaikkien osapuolten rajapintojen toimivan halutulla tavalla ja kehityksen voivan jatkua.

Toisessa vaiheessa koodattiin automatiikka sekä PHJ-järjestelmän API rajapinnassa olevaan integraatiomoduliin, että VHJ-järjestelmän DSP siirtoon. Tässä vaiheessa lisättiin myös aito viestin sisältö ja todennettiin sen siirtyvyys järjestelmien välillä.

PHJ:iin tulevat viestit tallennettiin omaan uuteen moduliin, johon luotiin attribuutit siirron sisällön mukaisesti ja määritettiin ulospäin siirrettävien elementtien osalta näkymät, joita siirto käyttäisi. Ulospäin siirtyvien laukaisimeksi määritettiin attribuutti jonka muuttaminen päälle aloittaisi siirron (Kuvio 8). Ensin arvo asetetaan arvoon To Be Integrated with VHJ, jolloin integraatio palvelu tunnistaa sen uudeksi siirrettäväksi elementiksi ja kun

ensimmäinen siirto on tapahtunut, muuttaa järjestelmä sen arvoon Integrated to VHJ. Tällöin myös järjestelmän käyttäjä voi tunnistaa, että integraatio on käytössä ja tiedot ovat siirtyneet ensimmäisen kerran ilman, että sitä pitää tarkistaa toisesta järjestelmästä.



Kuvio 8 Integraatio status määrittäminen

Päivitysten osalta koodattiin integraatiomoduli tarkistamaan elementtien, joiden Integraatio status oli aktiivinen, viimeisintä muutostietoa kymmenen minuutin sisällä. Jos siirtoon määritettyjä elementtejä löytyy, siirrettäisiin kyseisen elementin tiedot VHJ:iin.

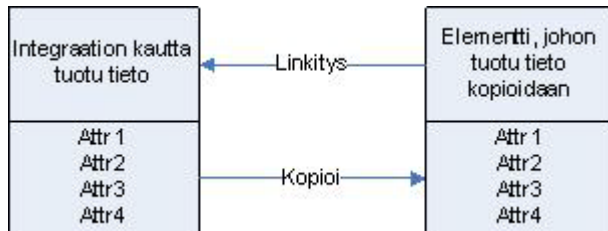
VHJ toimi muutoin vastaavasti, mutta sisään tuleva informaatio sijoitettaisiin olemassa olevan tyyppitiedon mukaisesti tietokantaan, ei erilliseen moduuliin. Lisäksi siirto tapahtuisi DPS serverin kautta API:n sijaan.

4.3 Tiedon käyttö järjestelmissä

Siirtoratkaisun toteuttamisen jälkeen ensimmäiset siirrot saatiin onnistuneesti suoritettua ja PHJ:n ja VHJ:n sisäisen osuuden kehittäminen voitiin aloittaa. Tiedot siis siirtyivät nyt järjestelmien välillä, mutta esittäminen molemmissa järjestelmissä piti vielä määritellä, että tiedot saatiin loppukäyttäjien käyttöön.

PHJ vastaanotti tiedot erilliseen integraatiomoduliin, johon jo aikaisempien integraatioiden toimesta tallennettiin toisista järjestelmistä vastaanotettua tietoa. Tiedon hyödyntäminen oli aikaisemmissa toteutuksissa tehty siten, että järjestelmän loppukäyttäjät liittivät tarpeelliset elementit järjestelmän muihin elementteihin. Tämä vaati linkki-tyyppisen attribuutin tekemistä niihin moduuleihin, joihin linkitys tahdottiin tehdä. Linkityksen myötä on mahdollista myös kopioida tietoja linkitettyjen elementtien välillä. Tällä tavalla voidaan

tehdä, esimerkiksi taulukkoja, joissa näytetään linkin takana olevia attribuutteja (Kuvio 9, Kuvio 10). Tietokantamielessä tämä ei kuitenkaan tarkoita tiedon periytymistä, vaan kirjaimellisesti kopiointia.



Kuvio 9 Linkitys PHJ:ssa

2. Time Schedules	
Link to Schedules	417-56153:BB79T Linkki
EM-4	-
EM-3	2010-02-26
EM-2	2010-01-19
EM-1	2010-01-19
EM0	2010-03-19
EM1	2010-04-09
EM2	2010-04-30
EM3	2010-05-14
EM4	2010-05-21
EM5	-
EM6	-
EM7	2010-09-17
EM8	2010-12-13
EM9	2011-02-25

Linkin takaa kopioidut päivämäärät

Kuvio 10 Linkityksen näkymä loppukäyttäjälle

Myös tässä uudessa tapauksessa päädyttiin samaan menetelmään, eli manuaaliseen linkkaamiseen, koska avain tietojen yhdistämiseen automaattisesti puuttui. Eduksi todettiin myös se, että menetelmä oli tuttu jo suurimmalle osaa käyttäjistä ja siten koulutus olisi helpompaa ja nopeampaa

VHJ toteutus tehtiin täysin vastaavasti. Tiedon siirto laukaistaan integraatio attribuutin aktivoimisella. Tässä tapauksessa se tehtiin lisäämällä valintapainike käyttöliittymään valintalistan sijaan. Käyttäjän merkityksessä valintapainike, siirsi järjestelmä valitun tiedon DSP serverille, joka työnsi sen edelleen AIP:iin. Sisään tuleva tieto otettiin käyttöön vastaavasti linkittämällä se oikeaan tietoon käyttäjän toimesta, kuten PHJ:ssakin.

4.4 Testaus

Kun asennukset oli saatu päätökseen molempien järjestelmien hiekkalaatikoissa, voitiin aloittaa käyttötapausten testaaminen. Testit perustuivat suoraan käyttötapausvaatimuksiin ja erillisiä testitapausedokumentteja ei tehty. Tähän työhön otti osaa koko projektitiimi siten, että kaikki osat tulivat testattua tulevien loppukäyttäjien näkökulmasta.

Testipäivän aikana kokeiltiin siirron toimivuutta erilaisin tavoin. Siirrettävään tietoon sotkettiin tarkoituksella duplikaatteja, väärän muotoisia arvoja ja tyhjiä rivejä. Ensimmäisten testien jälkeen luotiin virheloki, johon tuli kohtia mm. odottamattomista virheistä, joita tiedon sekoittaminen oli aiheuttanut.

Korjauskierroksen jälkeen tehtiin pikainen virhekäsittelytestaus kehitystiimien välillä. Testauksessa todettiin ensimmäisen testisession aikana kirjattujen virheiden olevan nyt kunnossa ja tämän jälkeen siirryttiin seuraavalle testausasolle, laadunvarmistusympäristöön. Tässä vaiheessa mukaan otettiin lisää loppukäyttäjien edustajia, koska tarkoitus oli samalla kouluttaa avainkäyttäjiä uuteen toiminnallisuuteen. Testaus eteni suunnitellusti, kunnes vastaan tuli kriittinen ongelma tiedon sisällössä.

Tämä ongelma ilmaantui kuitenkin poikkeuksellisesti siirrettävän tiedon lähteestä, joka paljastui olevan väärä VHJ:n päässä. Toisin sanoen siirrosta mukana oleva tieto ei ollut sitä, mitä oli tarkoitus siirtää. Tämän syyksi paljastui terminologinen ristiriita järjestelmien ja osallisten tiimien toimintamallien välillä. Kun määrittäsvaiheessa kaikki osallistuivat vaatimusten tekoon, puhuttiin samalla termillä kahdesta eri tietotyypistä ja tämä johti tilanteeseen, että

kehitystiimi koodasi siirron väärään kohtaan tietorakennetta. Tämä virheellinen lähde paljastui vasta toteutuksen testauksen yhteydessä, koska määräyksissä käytetty termi oli oikea kaikille asianosaisille.

Virheen korjaaminen ei ollut kooditasolla suuritöinen, mutta aiheutti aikataulullisesti viikon viivästymisen projektiin. Syynä oli se, että laadunvarmistusympäristöön ei voi tehdä korjauksia suoraan, vaan ne pitää ensin testata kehitysympäristössä ja tuoda ohjeiden mukaisesti alhaalta ylös. Lisäksi uusien pakettien asennus vaatii työpanosta tahoilta, jotka eivät olleet suoraan projektin alaisuudessa. Näiden tahojen resurssit saadaan käyttöön ainoastaan tilaamalla työ virallista reittiä, sovittujen prosessien mukaisesti.

Kolmannella testikerralla kaikki todettiin toimivan halutulla tavalla ja projektia voitiin esittää eteneväksi seuraavaan tasoon, eli siirtymistä tuotantoympäristöön. Viikon myöhästyminen aikataulussa aiheutti kuitenkin sen, että alkuperäiset työvaraukset olivat rauenneet. Tämä tehtävä oli projektipäällikön vastuulla, mutta yrityksen toimintatapoja tuntemattomana hän ei tiennyt käyttämättömien varausten raukeamisesta, mikäli niitä ei käytetä varatun aikajakson sisällä. Tästä koitui toinen viikon viivästyminen projektiin.

4.5 Käyttöönotto

Viivästyksistä huolimatta toteutus siirrettiin tuotannolliseen ympäristöön onnistuneesti. Koska avainkäyttäjät olivat osallistuneet testaukseen, oli toiminnallisuus heille tässä vaiheessa jo tuttu, mutta kommunikointiin tarvittava materiaali puuttui. Projektitiimi teki yhdessä kommunikointimateriaalin, jossa selitettiin yksityiskohtaisesti uusi toiminnallisuus askel askeleelta. Lisäksi tehtiin yhden sivun kokoinen pikaohje käyttöönoton tukemiseen. Tämän jälkeen avainkäyttäjät aloittivat omien vastuualueidensa henkilöstön kouluttamisen ja kommunikointia tuettiin sähköisellä uutiskirjeellä, joka lähetettiin kaikille käyttäjille.

Samana aikaisesti viimeisteltiin tekninen dokumentaatio palvelun toiminnallisuuksista. Tämä materiaali toimii tukena järjestelmien ylläpidosta

vastaaville tahoille ja sen valmistaa pääsääntöisesti koodauksesta vastanneet henkilöt yhdessä palveluiden teknisten ylläpitäjien kanssa. Näistä dokumenteista löytyy esimerkiksi järjestelmien rajapintojen määrittelyt, palvelimien osoitteet ja siirron logiikka. Tähän dokumentaatioon on myös COP:n mukaiset dokumenttipohjat, jossa määritellään kaikki vaadittavat tiedot.

Koska käyttäjämäärä on melko suuri ja hajaantunut laajalle maantieteelliselle alueelle, panostettiin kommunikointiin varsin paljon. Vaikka projekti ja sen tuoma muutos ei ollut kovin suuri, tai koskettanut läheskään kaikki käyttäjiä, haluttiin pitää huoli mahdollisimman kattavasta tiedottamisesta. Vastaanotto oli positiivinen ja uusi toiminnallisuus sai heti ensi vaiheessa kiitosta tuomastaan lisäarvosta. Myös toiveita jatkokehittämisestä ja konseptin laajentamisesta toisille alueille ilmaistiin useasta suunnasta.

4.6 Ylläpitovaiheeseen siirtyminen ja projektin päättäminen

Kun tuotantoympäristö oli saatu toimimaan, kaikki käyttäjät koulutettua ja dokumentaatio valmiiksi, voitiin palvelu siirtää normaaliin ylläpitovaiheeseen. Tässä kyseisessä tapauksessa mukaan ei tullut uusia toimijoita, joten kyse oli lähinnä lisäyksestä olemassa olevaan palveluportfolioon. Tässä vaiheessa ohjausryhmä päätti saavuttaneen E4 tason, jonka kriteerinä oli tuotannollisen palvelun olemassa olo ja ylläpitovaiheeseen siirtyminen.

E4 tason ja projektin sulkemiseen sovitun E5 tason väliin oli alunperin suunniteltu kuukauden aika, joka oli tarkoitettu mahdollisesti ilmaantuvien ongelmien ratkaisuun. Kahden viikon päästä E4 päätöksestä kuitenkin todettiin, että mitään ongelmia ei ollut ilmaantunut ja tätä varattua aikaa ei välttämättä tarvitse käyttää kokonaisuudessaan. Lisäksi testauksen aikana tapahtuneiden viivästysten vuoksi E4 päätös oli myöhästynyt kahdella viikolla. Tämän vuoksi katsottiin, että projekti voidaan sulkea alkuperäisen aikataulun mukaisesti, mikäli loppuraportti saadaan tuotettua ajoissa. Tässä haasteeksi muodostui resurssien siirtyminen projektista toisiin tehtäviin ja projektipäällikkö joutui keräämään palautteet pääasiassa sähköpostilla. Loppuraportti saatiin kuitenkin

tuotettua kohtuullisessa ajassa, koska projektin sisältö oli melko suppea. Projekti päättyi noin viikon alkuperäisestä aikataulusta myöhässä.

4.7 Projektin aikana opittua

Vaikka kyseinen hanke oli sisällöllisesti ja aikataulullisesti pieni, tuli sen aikana esille asioita, jotka koituvat usein kohtalokkaaksi ja estävät menestyksekkään projektin hallinnan. Kaikki alkaa tarpeen ymmärtämisestä ja jos tässä vaiheessa oletetaan asioita, johtaa se usein ei toivottuun lopputulokseen. Vaatimusten hallinnan merkitys on suuri, koska se toimii perustana kaikelle tekemiselle projektin aikana, aina projektin perustamispäätöksestä sen loppuun asti. Jos alussa tehdään virheellisiä johtopäätöksiä tai ymmärretään asioita väärin, voi se johtaa mittaviin aikataulullisiin ja taloudellisiin tappioihin.

Järjestelmäkehityksen hankkeet ovat pääsääntöisesti asiantuntijuutta vaativia ja niiden koordinointiinkin vaaditaan teknistä osaamista. Tämän vuoksi on riskialtista ottaa vetovastuuseen resurssi, jolla ei ole käsitystä toimintaympäristöstä tai tavasta, vaikka itse projektijohtaminen olisikin hallinnassa. Lyhyessä ja hektisessä kehityssyklissä ei ole yksinkertaisesti aikaa perehdyttää uusia resursseja ja tämä johtaa helposti väärinymmärryksiin ja huomiotta ottamisiin. Asiantuntija organisaatiossa oletetaan usein kaikkien tuntevan toimintatavat ja unohdetaan varmistaa, että tämä on myös totta. Lisäksi avainresurssit on saatava sitoutumaan koko projektin ajalle, ottaen huomioon mahdolliset aikataululliset haasteet ja myöhästymiset.

5 Loppulause

Integraatiokehitys on haasteellinen ja jatkuvasti kasvava osa yritysten tietojärjestelmäkehityksen kokonaisuutta. Uusien, käyttöön otettavien järjestelmien ja olemassa olevien palveluiden pitää pystyä vaihtamaan tietoa keskenään siten, että se on ajantasaista kaikissa niissä paikoissa, joissa sitä näytetään. Tiedon tallentaminen halutaan keskittää mieluiten yhteen järjestelmään, josta sitä tarpeen mukaan jaetaan toisiin. Tällä tavalla vähennetään inhimillisten erehdyksien määrää ja helpotetaan käyttäjien työkuormaa.

Teknisten ratkaisujen kehittyessä järjestelmien yhteensovittaminen käy helpommaksi, mutta edelleen haasteita riittää itse tiedon muodossa ja yhteensopivuudessa. Tämän vuoksi arkkitehtuurista suunnittelua ei saa unohtaa ja sen rooli itse asiassa vain kasvaa informaation määrän lisääntyessä. Vastaavasti paineet nopeaan muutoskyvykkyyteen ovat vastavaatimuksena tiukalle arkkitehtuurille, joka usein realisoituu tiukkana, prosessisidonnaisena pakottamisena tiettyyn toimintamalliin. Toimintatavat ja itse hallittava tieto muuttuu ja kehittyy niin nopeasti, että järjestelmien pitää pystyä mukautumaan joustavasti uusiin tilanteisiin.

Toimivat ja luotettavat tiedonsiirtomekanismit ja -palvelut tuottavat lisäarvoa yrityksille parempana laatuna niin tiedossa, toiminnoissa kuin tuotteissa sekä parantavat tuottavuutta työntekijöiltä vapautuvana työaikana.

6 Lähteet

Markus Valo, System Design Architect. Haastattelu 12.4.2009. Mobiiliteknologia yritys.

COP, System Design Guidelines. IT sisäinen ohjeistus 12.4.2009. Mobiiliteknologia yritys.

N.N.(1) 5.03.2009. Projektipäällikön haastattelu. Mobiiliteknologia yritys

N.N.(2) 4.02.2009. PHJ LCO haastattelu. Mobiiliteknologia yritys

N.N.(1) 6.02.2009. VHJ LCO haastattelu. Mobiiliteknologia yritys

7 LIITTEET

7.1 Käyttötapausesimerkki

Sample date integration (Display module)

Use Case

Related Feature
 Related Requirement
 Objects
 Item Code
 Originator Petri Havia
 Function D R&D TEP
 Approver
 Document ID
 Status Approved
 Location DocLib

Change History

Version	Date	Handled by	Reason for Change
1.0	26-Jan-2009	Petri Havia	First draft

Table of Contents

1. INTRODUCTION	34
-----------------	----

1.1 Scope	34
1.2 Roles / Access level	34
1.3 Assumptions	34
2. BASIC FLOW OF EVENTS	35
3. ALTERNATIVE PATHS	36
3.1 [Actore Creates new catalogue item (separate use case)]	
4. RELATED BUSINESS RULES	36
5. EXCEPTION PATHS	36
5.1 {No access to system or access level restricted}	36
6. APPENDIX	36
6.1 Glossary	

8 Introduction

This use case describes integration functionality between PHJ and VHJ applications in sample date synchronization use case.

Sample date information is set of four attributes in PHJ including Description, Plan Date, Updated Plan Date and Actual Date. These attributes are gathered under heading called Sample Availability Infocamation

8.1 Scope

- This Use Case concerns in PHJ Display Module Elements and related views

8.2 Roles / Access level

User

Role	Access to Object Types	Notes
Element owner	Person who owns element in PHJ and is aallowed to edit and release this information to other systems, in this case VHJ	

8.3 Assumptions

The Use Case is written supposing that

- All needed access rights exists
- Needed member groups exist
- Person is connected to group in case and
- Member group has edit view in use

9 Basic Flow of Events

[Square Brackets] denote links to alternative paths. {Curly Brackets} denote links to exception paths. These alternatives are described in own sections in this document.

1. Actor opens PHJ application

System: Opens PHJ UI

Exception: {No access to system or access level restricted}

Alternative:

2. Actor selects work space from available work space list (Chipset, CWS & AV HW Portfolios)

System: Opens selected work space

Exception: {No access to system or access level restricted}

Alternative:

3. Actor selects 'Display' from left side menu and selects view that allows to modify display related information

System: Opens selected view (latest user selection is displayed)

Exception:

Alternative:

4. Actor selects Element from tree view or table view

System: Stores Select Currency value

Exception:

Alternative:

5. Actor opens attribute to edit mode and fills relevant sample date information to attribute(s)

System: Opens edit window

Exception:

Alternative:

6. Actor saves made changes

System: Stores filled attribute information and records history information about attribute

Exception:

7. Actor opens Integration status attribute and selects value 'To be integrated with VHJ'

System: Stores filled attribute information and records history information about attribute. Starts integration activities.

Exception:

Alternative:

8. System moves sample date information to VHJ thru AIP

System: Uses needed transactions to move information

Exception:

Alternative:

9. System receives information to VHJ

System: Information is stored to system

Exception:

Alternative:

10. Actor takes sample date information in use in VHJ by connecting document and imported information together

System: Links information in system and displays integrated information in selected document

Exception:

Alternative:

10 Alternative Paths

Alternative path describes other possibility in decision point. Alternativity can be also provided by system according to related business rule.

11 Related Business Rules

One negative answer to the business rules is enough to prevent operation

12 Exception Paths

12.1 {No access to system or access level restricted}

User is not authorized to do selected operation in system

13 Appendix

Views are defined in separate documents that can be found from location: