

OULUN SEUDUN
AMMATTIKORKEAKOULU



Jarkko Ruottinen

ETÄKUNTOUTUSJÄRJESTELMÄN SUUNNITTELU JA KEHITYS

ETÄKUNTOUTUSJÄRJESTELMÄN SUUNNITTELU JA KEHITYS

Jarkko Ruottinen
Opinnäytetyö
Syksy 2013
Tietotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, ohjelmistokehitys

Tekijä(t): Jarkko Ruottinen

Opinnäytetyön nimi: Etäkuntoutusjärjestelmän suunnittelu ja kehitys

Työn ohjaaja(t): Lasse Haverinen (OAMK) ja Vesa Kajanus (BonWell Intelligence Oy)

Työn valmistumislukukausi ja -vuosi: Syksy 2013

Sivumäärä: 31

Tuoreessa oululaisessa yrityksessä kehitetään järjestelmää, joka tarjoaa asiakkaalle etäkuntoutuspalveluja eri muodoissa. Tuotekonseptissa käytetään yrityksen omia mittausjärjestelmiä, joilla mahdollistetaan yksilöllinen kuntouttaminen ja kuntoilu. Harjoitteet toteutetaan pelillistämällä. Toimeksiantaja tälle opinnäytetyölle oli BonWell Intelligence Oy Ltd.

Työn tavoitteena oli selvittää peliharjoitteen toteuttamiseen liittyvät tarpeet etäkuntoutusjärjestelmästä sekä ohjelmoida peliharjoite, joka perustuu järjestelmäselvitykseen. Työ toteutettiin tuotekonseptiin, jossa on erilaisia sulautettuja järjestelmiä ja ohjelmistoja. Työssä selvitettiin järjestelmävaatimuksia peliharjoitteelle sekä peliharjoitteen vaatimuksia järjestelmälle.

Yritys käyttää opinnäytetyössä tehtyjä järjestelmänmäärittämiä, havaintoja sekä yleisiä toimintamalleja tuotteen jatkokehittämisen suuntaviivoina. Järjestelmämäärittäykset mahdollistavat pelisuunnittelun teknisen rajauksen. Ohjelmointityönä ei toteutettu peliharjoitetta, vaan työkalu helpottamaan mittauksien analysointia ja käsittelyä yrityksen päivittäisessä työssä.

Asiasanat: pelillistäminen, peliteollisuus, järjestelmäsuunnittelu, ohjelmointi

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software Development

Author(s): Jarkko Ruottinen

Title of thesis: Design and development of remote rehabilitation system

Supervisor(s): Lasse Haverinen (OUAS) ja Vesa Kajanus (BonWell Intelligence Oy)

Term and year when the thesis was submitted: Autumn 2013 Pages: 31

Fresh Oulu-based Company is developing system that provides remote rehabilitation services in various forms to customers. The product concept is using company's own measurement system, what enables individual rehabilitation and exercise training. Exercises are carried out with gamifying. Employer for this thesis work was BonWell Intelligense Oy Ltd.

Objective of this work was to find system definitions of gamified rehabilitation training and program game demo using those system definitions. Thesis work was implemented in product concept, where was different types of embedded systems and software's. Thesis work researches system demands to gamified training as well as gamified training demands to the system.

The company uses thesis work's system specifications, observations, and the functional models as guidelines for further development of the product. System specifications allow for scoping technical design of the game. Game demo was not implemented, instead of that, there was produced tool that facilitates the analysis and processing of the measurements at company's day to day work

Keywords: gamify, game industry, system design, programming

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
SANASTO	6
1 JOHDANTO	7
2 SUUNNITTELUN AIHEALUEET JA PROSESSIT	8
2.1 Pelisuunnittelu	8
2.2 Ohjelmisto- prosessi ja tekniikka	8
2.3 Ongelmanratkaisumalli	10
2.4 Järjestelmäsuunnittelu	11
3 JÄRJESTELMÄN SUUNNITTELU	14
3.1 Lähtötiedot	14
3.2 Suunnittelun pelillinen näkökulma	15
3.3 Järjestelmät, ohjelmistot ja työkalut	18
4 OHJELMISTON TOTEUTUS	19
5 PÄÄTELMÄT JA JATKOKEHITTÄMINEN	22
5.1 Prosessit ja toiminta	22
5.2 Ohjelmistojen kokoonpano ja hallinta	23
5.3 Pelillisuus ja harjoitteet	25
5.4 Järjestelmän hallinta	29
6 LOPPUSANAT	30
LÄHTEET	31

SANASTO

KPA	Key Process Area eli keskeinen prosessialue
SEI	Software Engineering Institute eli ohjelmistokehitys instituutti
HTML	Hypertext Markup Language
BT	Bluetooth
WAV	äänitiedostomuoto
CSV	tekstimuotoinen tiedostomuoto, johon tallennetaan taulukkotietoa

1 JOHDANTO

Tällä hetkellä globaalit terveyspelien markkinat ovat vasta syntymässä ja alan ilmiötä kuvaava terminologia on vielä hahmottomatonta. Terveyspelien markkinoissa on potentiaalia, mutta se ei ole ainoa syy, miksi huomattava määrä kehittäjiä keskittyy hyötypeleihin, erityisesti terveyspeleihin. (1, s.12.)

BonWell Intelligence on oululainen start up -yritys, joka on perustettu tammi-kuussa 2013 ja jonka toimialana on hyvinvointi ja sen pelillistäminen. Yritys haluaa tuoda pelillistämisen avulla mielenkiintoa ja hauskuutta hyvinvointisovelluksiin. (2)

Hyvinvointisovellus mahdollistaa täysin uuden ja joustavan tavan tuottaa kuntoutuspalveluja niitä tarvitseville ajasta ja paikasta riippumatta. Pelillistämällä tuodaan tuloksellisuutta ja vaikuttavuutta kotikuntoutukseen, jonka kuntoutuja haluaa suorittaa annettujen ohjeiden mukaan. (2)

Opinnäytetyön aiheen sain BonWell Intelligence Oy:ltä, joka halusi tutkia etäkuntoutuksen toteuttamista pelillistämällä omassa tuotekonseptissaan. Tässä opinnäytetyössä selvitetään järjestelmävaatimuksia pelillistämisen näkökulmasta.

Työn pääasiallisina tavoitteina oli selvittää pelillisen harjoitteen toteuttamiseen liittyvät tarpeet järjestelmästä ja saada mahdollisimman selkeä kuva järjestelmän tarpeisiin pelillisten harjoitteiden toteuttamista varten. Työhön kuuluu ohjelmointityö.

2 SUUNNITTELUN AIHEALUEET JA PROSESSIT

Start up -yritysten prosessit auttavat yritystä varmistamaan toimintojen jatkuvuutta, luomaan omistajuuden ja osallistumisen kulttuuria, muutosten hallintaa ja kasvattamaan ennustettavuutta. Lisäksi prosessit auttavat yritystä saavuttamaan tunnustusta. (3, s.5.)

2.1 Pelisuunnittelu

Pelisuunnitteluprosessissa käytetään pelisuunnitteludokumenttia. Suunnittelu-dokumentissa kuvataan pelin tuottamisprosessia, jossa keskitytään pääsääntöisesti pelin sisältöön ja pelin toteutukseen liittyviin ominaisuuksiin. Yleisesti dokumentissa kuvataan peliprojektin kulku pelimääritykset ja rahanansaintamalli. Pelin rajaus määritetään laitteistotason rajoitteilla, joita ovat yleisellä tasolla käyttöjärjestelmät, laitteet sekä käyttöympäristö. (4, s.1-4.)

2.2 Ohjelmistoprosessi ja -tekniikka

Ohjelmistotekniikka on kerrosrakenteista teknologiaa. Sen kivijalkana toimivat prosessikerrokset (kuva1).



KUVA 1. Ohjelmistoprosessin tasot (5, s.19.)

Yleisellä tasolla tekniikka koostuu analyysistä, suunnittelusta, rakennelmasta, vahvistamisesta ja teknisen kokonaisuuden hallinnasta.

Tekniikan aiheen vaikuttamatta seuraaviin kysymyksiin täytyy löytää vastaus:

- Mikä ratkaistava ongelma on?
- Millaisia kokonaisuuden ominaisuuksia on käytetty ongelman ratkaisuun?
- Miten kokonaisuus ja ratkaisu toteutetaan?
- Millaisia tapoja käytetään havaitsemaan suunnittelun ja kokonaisuuden virheet?
- Miten kokonaisuuden pitkänajanhallinta toteutetaan, kun kokonaisuuden käyttäjä pyytää korjauksia, mukauttamisia ja parannuksia?

SEI (Software Engineering Institute) on kehittänyt ohjelmiston kypsyysmallin, jossa organisaatioiden ohjelmistokehityksen kypsyyttä voidaan arvioida. Tasoina ovat varhainen, toistettava, määritetty, hallittu ja tehostava. Malliin on määritetty kahdeksantoista keskeistä prosessialuetta eli KPA:t. Varhaisessa tasossa ei ole määritettyjä KPA:ta, koska toiminnallisuudet perustuvat yksilöihin ja muutamisiin määritettyihin prosesseihin.

Prosessin kypsyystaso 2 sisältää:

- ohjelmistokokoonpanohallinnan
- ohjelmistolaatuhallinnan
- ohjelmistoalihankintahallinnan
- ohjelmistoprojektin seurannan ja valvonnan
- vaatimustenhallinnan.

Prosessin kypsyystaso 3 sisältää:

- vertaisarvioinnin
- työryhmäkoordinoinnin
- ohjelmistotuotekehityksen
- integroidun ohjelmistonhallinnan
- koulutusohjelman
- organisaatioiden prosessimäärittelyt
- organisaation prosessien painopisteet.

Prosessin kypsyystaso 4 sisältää:

- ohjelmiston laadunhallinnan
- määrällisen prosessinhallinnan.

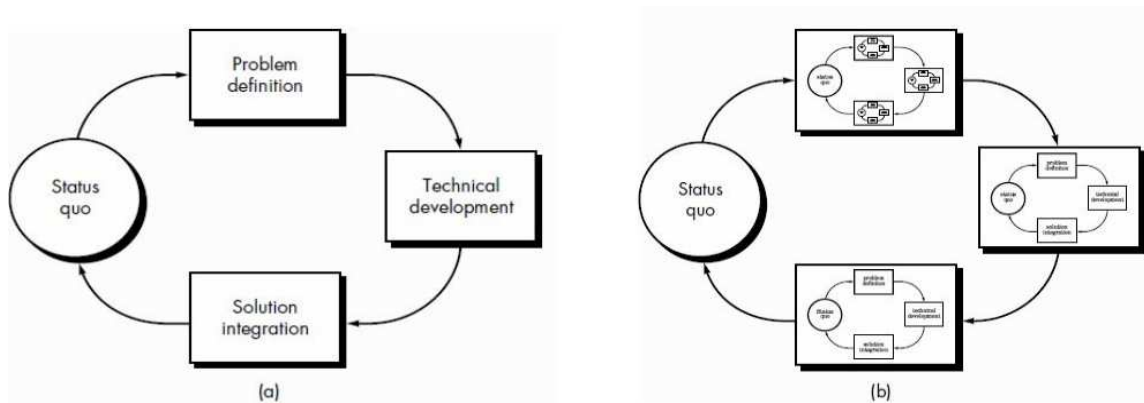
Prosessin kypsyystaso 5 sisältää:

- prosessien muutoshallinnan
- teknologian muutoshallinnan
- vikojen ehkäisyyn.

Jokaiseen KPA:han on määritetty avainkäytännöt, jotka edistävät päämäärien saavuttamisen. Avainkäytäntöjä ovat politiikat, menettelyt ja toiminnot.

2.3 Ongelmanratkaisumalli

Lineaarinen sekvenssimalli on systemaattisesti ja peräkkäisesti lähestyvä tapa ohjelmistokehitykseen, joka alkaa järjestelmätasolta ja käsittelee analyysin, suunnittelun, ohjelmoinnin testauksen ja ylläpidon (kuva 2).



KUVA 2. Lineaarinen sekvenssimalli ongelmanratkaisuun ohjelmistokehityksessä. (5, s. 27.)

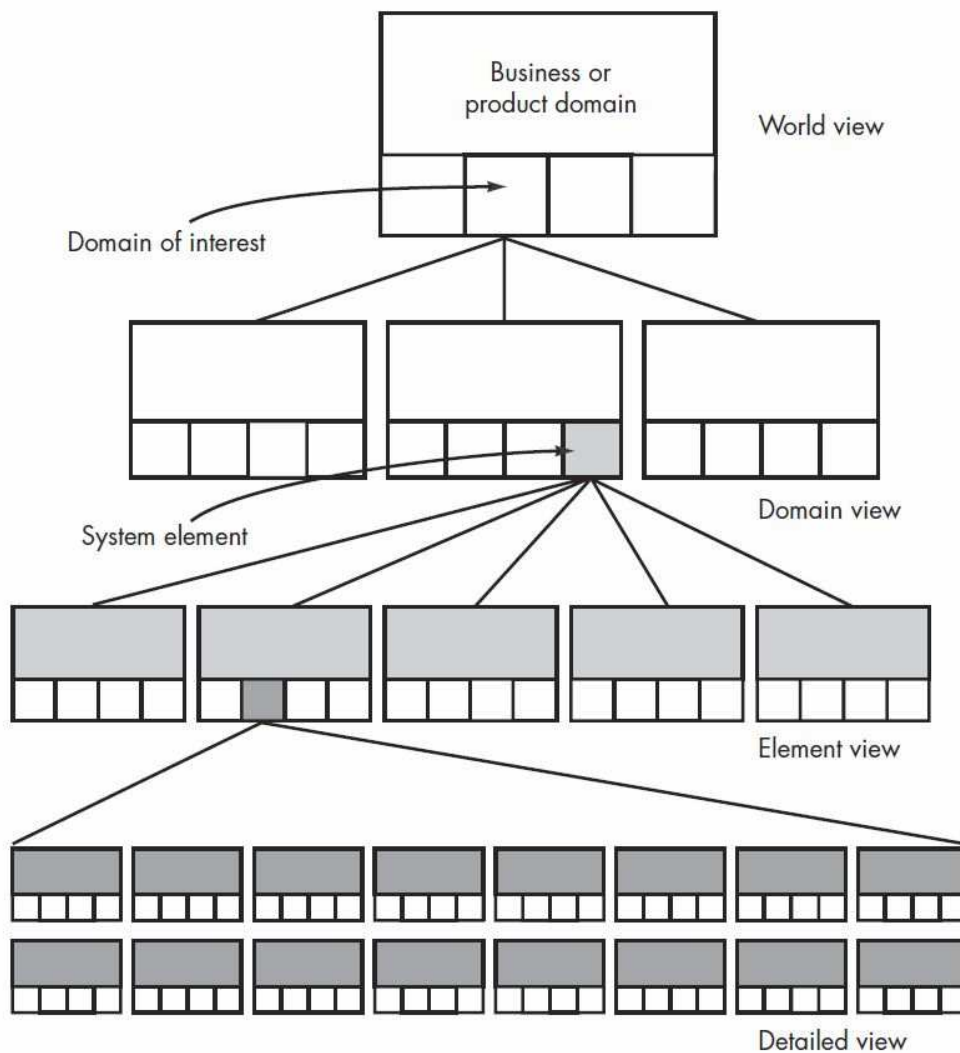
Koska ohjelma on aina osa järjestelmää tai liiketoimintaa, työ alkaa järjestelmän vaativuusmäärittelystä. Järjestelmänäkymä on oleellinen, kun ohjelmisto liittyy muihin elementteihin kuten laitteistoon, käyttäjiin ja tietokantoihin. Järjestelmäkehitys ja analyysi sisältävät järjestelmätason vaativuuden pienistä kokonai-

suuksista korkean tason suunnitteluun ja analyysiin. Tietotekniikan kehitys käsittelee vaatimusten keräämisen liiketoimintaan ja liiketoiminta-alueelle. (5, s.17-31.)

2.4 Järjestelmäsuunnittelu

Tietokoneisiin pohjautuvien järjestelmien tavoite voi olla kehittää liiketoiminta-alueita tai tuottaa myytävä tuote. Tietokoneisiin pohjautuvat järjestelmät käyttävät useampia järjestelmän osa-alueita tavoitteen saavuttamiseen kuten ohjelmistoa, laitteistoa, käyttäjiä, tietokantoja, dokumentaatiota ja toimintatapoja.

Tällaiset järjestelmät muodostuvat monimutkaisiksi, koska järjestelmän yksi osa-alue voi koostua makroalueesta, joka koostuu isommasta järjestelmästä (kuva 3).



KUVA 3. Järjestelmän hierarkia. (5, s. 244.)

Järjestelmäkehitys on mallinnusprosessi. Riippumatta keskittykö malli yleiseen vai tarkkaan näkymään, kehittäjä luo mallin, joka esittää:

- prosessimäärityksen, joka palvelee käsiteltävän asian tarpeita
- prosessin käyttäytymisen ja olettamukset, johon käyttäytyminen perustuu
- selkeästi ulkoiset ja sisäiset syötteet
- kaikki yhteydet, jotka auttavat ymmärtämään näkymää.

Vaatimusten kerääminen vaikuttaa yksinkertaiselta. Siinä kysytään asiakkaalta, käyttäjiltä ja muilta, mitkä ovat järjestelmän tai tuotteen päämäärät, mitä pitää

saavuttaa, miten järjestelmä tai tuote soveltuu tarpeisiin tai liiketoimintaan ja miten sitä käytetään päivittäin. Se ei ole yksinkertaista vaan todella vaikeaa.

Vaatimusten keräämisen ongelmia ovat toimintalaajuus, asiakkaiden tai käyttäjien ymmärtäminen ja epävakaisuus. Työn tuotoksen seurauksena kartoittamiin käytettävät aktiviteetit vaihtelevat, joihin vaikuttavat järjestelmä tai tuote, joka on rakennettu. Useimmissa järjestelmissä työn tuotteita ovat:

- Lausunto tarpeesta ja toteutettavuudesta.
- Rajattu selvitys järjestelmän tai tuotteen tavoitteesta.
- Luettelo asiakkaista, käyttäjistä ja muista sidosryhmistä, jotka osallistuvat vaatimusten määrittelytoimintoihin.
- Kuvaus järjestelmän teknisestä ympäristöstä.
- Luettelo vaatimuksista ja verkkorajoituksista, jotka koskevat kaikkia.
- Käyttötapauksista, jotka luovat näkemyksen järjestelmän tai tuotteen käyttöön erilaisissa käyttötapauksissa.
- Prototyypit, jotka on kehitetty helpottamaan vaatimusten määrittelyä.

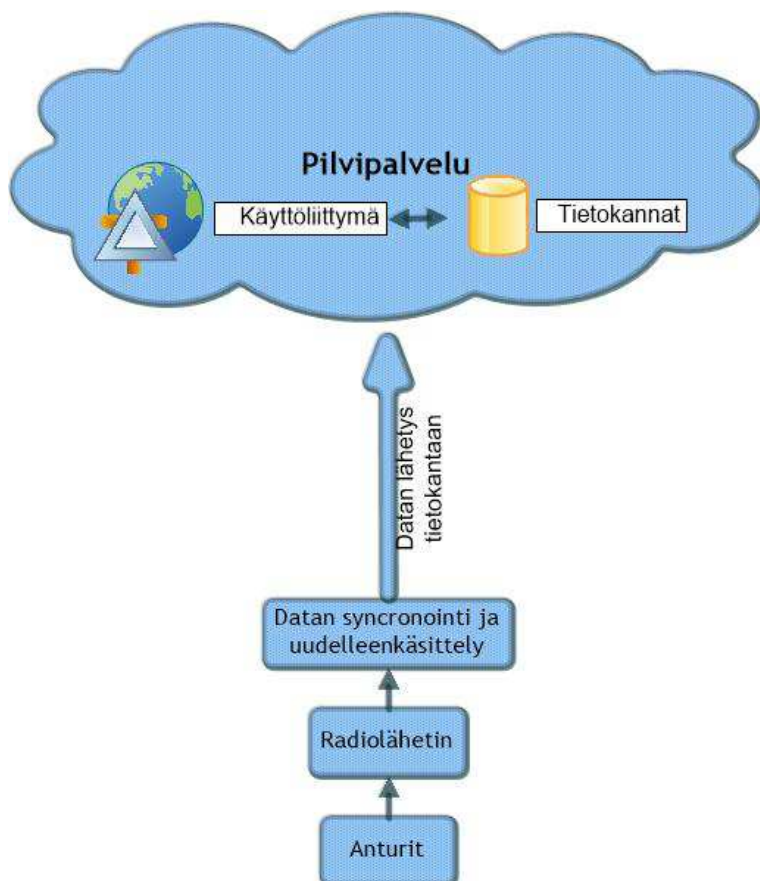
Järjestelmäkehitys vaatii tiivistä kommunikointia asiakkaan ja järjestelmäsuunnittelijan välillä. Tämä saavutetaan vaatimustenmäärittelytoiminnoilla, joita ovat koostaminen, analyysi, neuvottelu, määrittely, mallinnus, validointi ja hallinta. (5, s.241-265.)

3 JÄRJESTELMÄN SUUNNITTELU

Työ alkoi järjestelmään perehtymisellä, jonka aikana keräsin suunnitteluun tarvittavaa lähtötietoa. Järjestelmä on kokonaisuudessaan laaja, ja sen yhtenä liiketoiminnallisena päämääränä on saada käyttäjille pelillistä toimintaa. Järjestelmää toteutetaan palvelumuotoisena.

3.1 Lähtötiedot

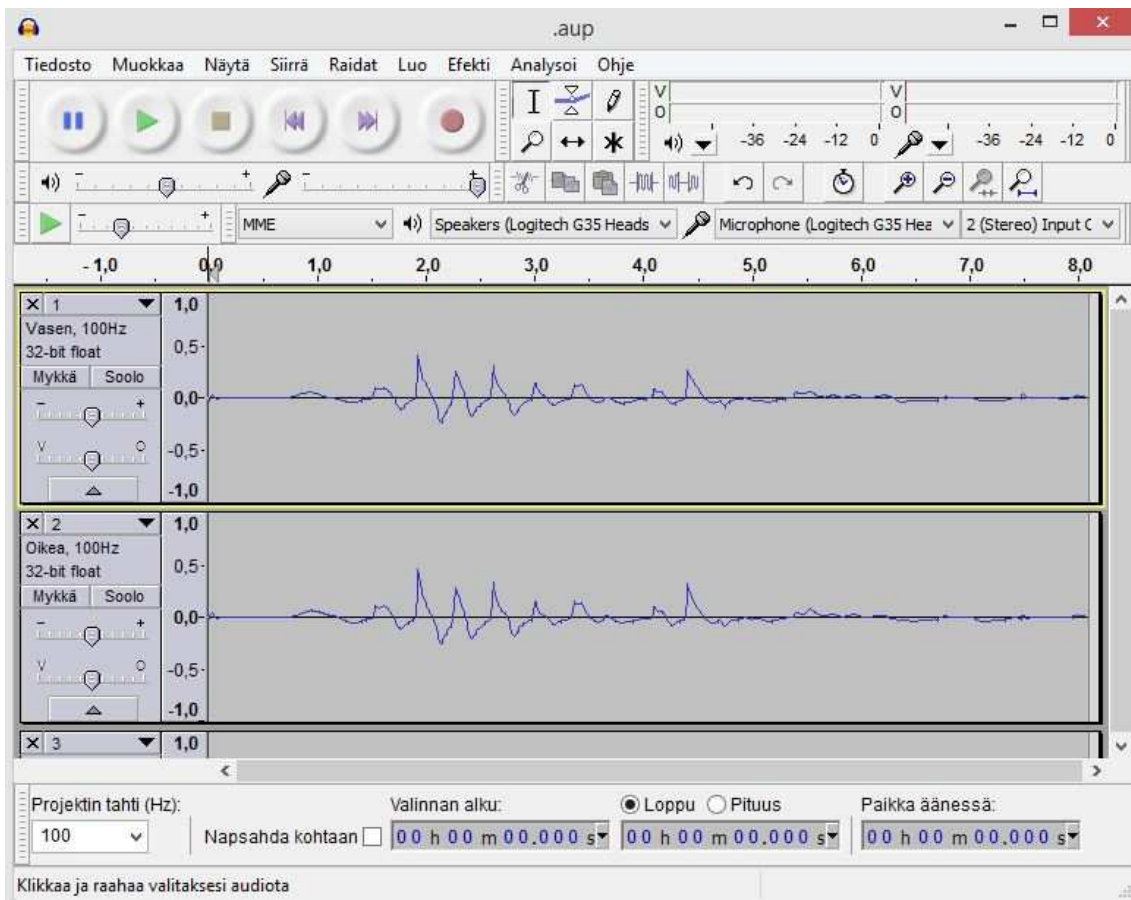
Suunnittelun lähtökohtana oli järjestelmä, jossa anturilaitteistolla kerätään mittaustietoa käyttäjän jaloista. Anturilaitteisto mittaa ja käsittelee mittaustuloksen radiolähteykseen. Mittaustulokset toimitetaan pilvipalvelun tietokantapalvelimeen. Tieto kulkee kolmannen osapuolen järjestelmän kautta. Järjestelmässä tieto muotoillaan ennen tietokantapalvelimeen lähettämistä. (Kuva 4.)



KUVA 4. Lähtötilanne järjestelmäkuvauksesta

Järjestelmästä on malleja eri käyttötapauksiin liittyvistä käyttöliittymistä ja tietokantarakenteista. Itse mittaustuloksista on oma tietokanta, jonka käyttöliittymä on rakenteilla.

Mittaustuloksia pystyy tallentamaan tiedostomuotoon kolmannen osapuolen palvelusta, jossa tiedostomuoto on wav-tiedosto. Tiedostoa käytetään Audacity-ohjelmalla (kuva 5.), jossa se saadaan graafiseen näkymään. Ohjelmasta tieto voidaan siirtää numeeriseen muotoon.



KUVA 5. Anturitietoa Audacity-ohjelmassa.

3.2 Suunnittelun pelillinen näkökulma

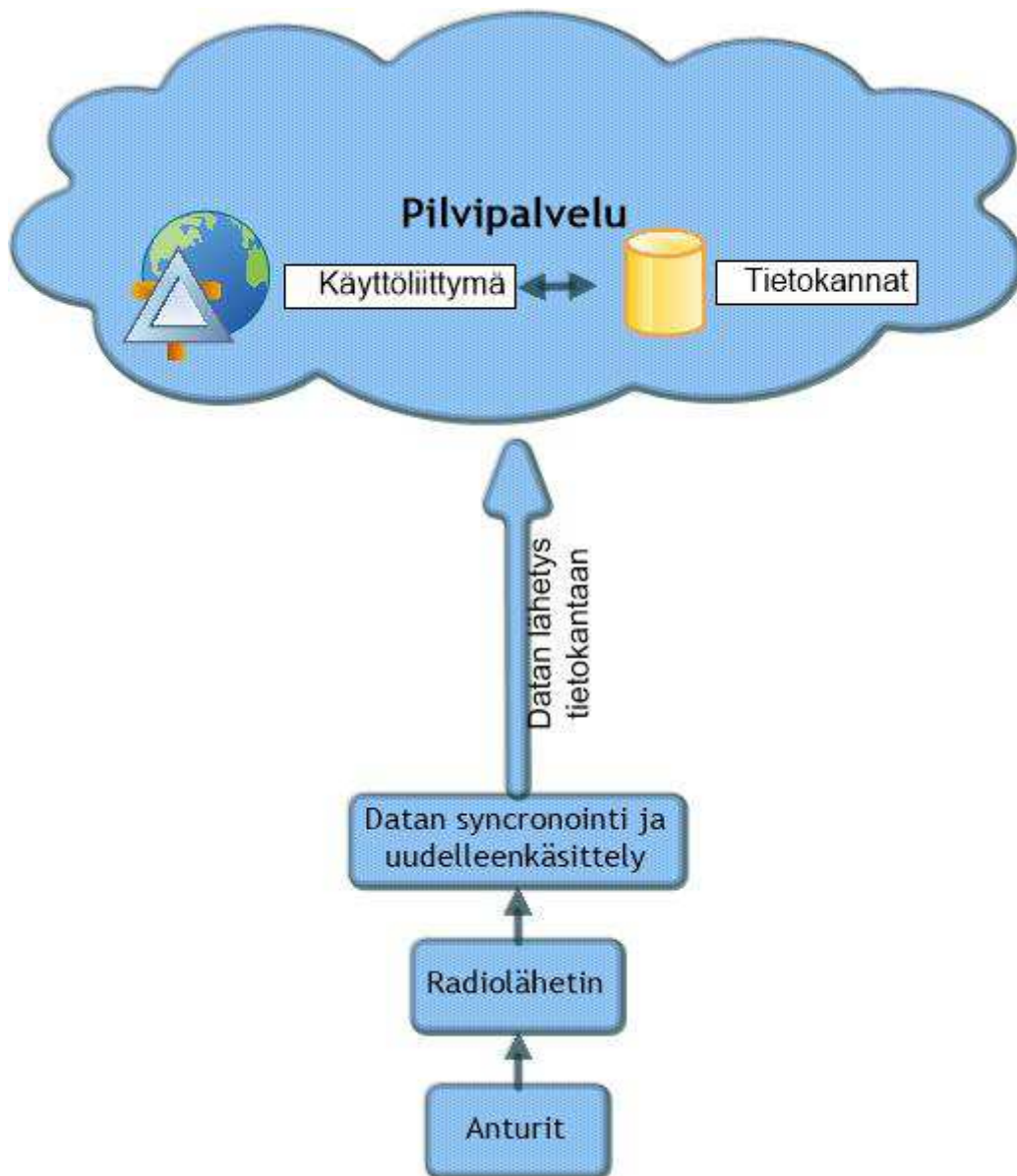
Tuotteena on palvelukonsepti, jossa käyttäjä tekee urheilullisia harjoitteita. Osa harjoitteista tehdään pelilliseen muotoon. Mittalaitteistoa käytetään ohjaimena,

joka toimii samanaikaisesti tietoa keräävänä laitteistona. Tietoa tallennetaan pilvipalvelun tietokantapalvelimiin, joista tietoa voidaan käsitellä jälkikäteen. Palvelukonseptiin liittyy, myös kolmannen osapuolen käyttöliittymät, joissa voidaan tehdä erityyppisiä analyysejä tuloksien ja henkilön historiatietoprofiiliin kautta.

Peliharjoitteista tehtiin mielikuvamatkoja mahdollisiin pelityyppeihin, joita nykyisillä antureilla voisi toteuttaa. Tämän jälkeen järjestelmää katselmoitiin peliharjoitteen teknisellä määrittelyllä.

Teknisen määrittelyn arvioinnissa havaittiin, että ohjaimen pitää olla riittävän reaaliaikainen, jotta voidaan toteuttaa toiminnallisesti reaaliaikaisia peliharjoitteita. Valmiiksi tehdyn radiokokoonpanon vasteaika selvitettiin. Radiokokoonpanossa mittauksesta tietokantapalvelimeen vasteaika oli pahimmillaan useita sekunteja, ja se vaihteli paljon. Sekuntien vasteaika rajoittaa peliharjoitteen muistipelityypiksi, joka ei liiketoiminnallisesti tässä vaiheessa ole tarkoituksenmukaista. Radiokokoonpanoa voidaan käyttää erityyppisissä harjoitteissa ja mittauksissa.

Riittäväksi vasteajaksi arvioitiin alle 100 ms. mittauksesta käyttöliittymään. Tällainen vasteaika mahdollistaa toiminnallisen harjoitteen, jossa palaute on riittävän reaaliaikainen käyttäjälle ja käyttäjä pystyy käyttämään laitteistoa ohjaimena. Riittävän pienen vasteajan saavuttamiseksi mittauslaitteistona käytetään työn alla olevaa Bluetooth-laitteistoa, jota kutsutaan BT-kokoonpanoksi (kuva 6).



Kuva 6. Järjestelmä BT-kokoonpanolla.

Riittävän vasteajan saavuttamiseksi pelillisen harjoitteen suunnittelussa pitää huomioida tarpeellinen tiedonkäsittely järjestelmän eri osioissa. Käyttöliittymänä voi olla HTML- kieleen perustuva selainratkaisu tai käyttäjän laitteeseen asennettava ohjelmisto. Palveluun liittyen kumpaakin vaihtoehtoa käytetään eri toiminnoissa.

BT-kokoonpanon ohjelmisto toteutetaan käsittelemään mittatietoa mahdollisimman suoraan käyttöliittymän toiminnallisuuteen liittyen, jotta vasteaika saadaan minimoitua. Lisäksi ohjelmiston pitää hoitaa tiedonsiirto tietokantaan.

3.3 Järjestelmät, ohjelmistot ja työkalut

Tuotekonsepti on kokonaisuudessa monimutkainen ja koostuu useista eri osaluista ja teknologioista. Yleisesti ottaen järjestelmässä on pilvipalvelut, asiakasohjelmistot ja mittauslaitteistot.

Pilvipalvelussa on tietokantoja ja internet-sivustoja. Internet-sivustot voivat sisältää asiakashallintaa ja harjoitteita, joita voidaan käyttää usealla eri teknologiaalustalla. Sivustoissa on useita eri osa-alueita, joissa käsitellään asiakkuuteen liittyviä henkilötietoja ja mittaustietoa.

Asiakasohjelmistot toteutetaan eri teknologia-alustoille omina toteutuksinaan, jotka voivat olla harjoitteita ja asiakashallintaan liittyviä ohjelmistoja. Järjestelmään liittyvät mittauslaitteistot ovat omia kokoonpanoja, jotka tekevät mittauksen ja lähettävät mittaustulokset järjestelmään.

Työkaluiksi valittiin Microsoftin tuotteet, joita pyritään käyttämään mahdollisimman monessa toiminnossa, jotta vältetään erilaisilta yhteensopivuusongelmilta. Työkalujen valinta vaikuttaa myös järjestelmän toteutukseen, koska paras yhteensopivuus ja tuki saadaan saman toimittajan tuotteista.

Peliharjoitteiden tekemiseen käytetään Unity 3D -pelimoottoria tai MS Visual Studio 2012 -ohjelmistoa. Ohjelmointikielenä käytetään C# -kieltä.

Valinnat ovat viitteellisiä, koska laitteistotason ohjelmoinnissa ei välttämättä ole tukea näille ohjelmistoille. Laitteistotason ohjelmointiin liittyvät ohjelmistot tarkastetaan laitteiston mukaisesti. Järjestelmätasolla tämä antaa suuntaa myös pilvipalvelun tuottajan valintaan sekä ensisijaisen asiakasohjelmiston järjestelmien valinnasta.

4 OHJELMISTON TOTEUTUS

Alkuperäisenä ajatuksena oli toteuttaa visuaalinen pelillinen demo, jossa mittautustietoa esitetään graafisesti. Toteutus muutettiin erilaiseksi, koska liiketoiminnallisesti ei ollut järkevää toteuttaa pelillistä demoa, joka ei kuitenkaan kytkeydy vielä järjestelmään ja näytä mittautustietoa riittävän reaaliaikaisesti.

Toteutuksen asiakkaaksi määräytyi yrityksen oma tarve. Tarpeena oli kehittää mittautuloksien analysointia ja tiedonkäsittelyä auttava työkalu, jonka ohjelmisto-osia voidaan hyödyntää järjestelmäkehityksessä myöhemmin. Ohjelmointityö toteutettiin radiokokoonpanolla.

Demovaiheen pilvipalvelu on toteutettu Google Cloud Platform -palvelussa. Prototyyppi on kolmannen osapuolen tuottama. Ohjelmisto toteutettiin osio kerrallaan, mikä aloitettiin yhteyden muodostamisella tietokantaan.

Ohjelmointi aloitettiin konsoliohjelmana, jossa otettiin yhteys palvelimeen. Kun yhteys oli muodostettu, jatkettiin sisään kirjautumisella, jossa lähetettiin sisään kirjautumistiedot sekä anturin tunnistuskoodi. Tämän jälkeen palvelin palautti mittautustiedot.

Mittautustiedot olivat XML-muodossa, ja palvelin palautti kaikki kyseisien anturien mittautustiedot. Mittaus ei siis ollut mitenkään helppolukuista ja sitä oli paljon. Tästä huomiosta teetätimme muutoksen palvelimeen, jossa mittautuloksia vähennettiin. Tämä asia huomioidaan myöhemmin palvelimeen tehtävässä ohjelmointirajapinnassa.

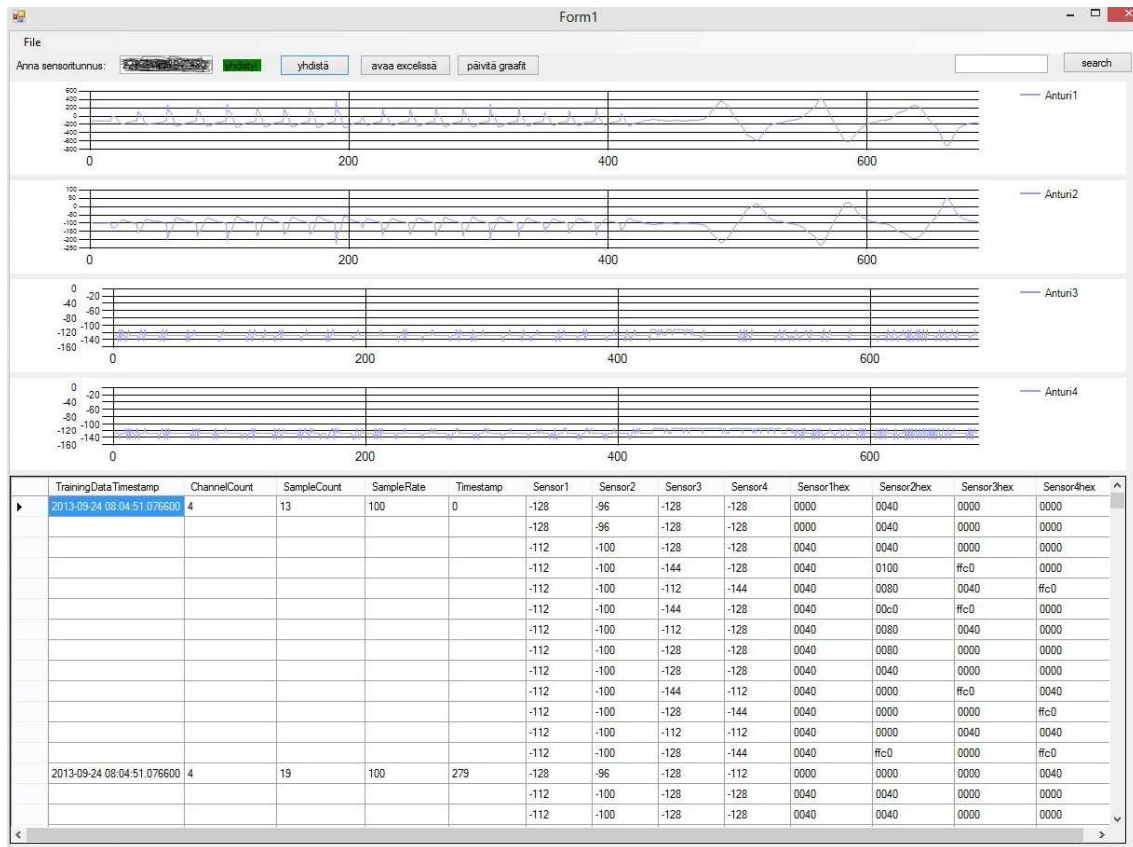
Kun mittautulosten syöte oli jäsennelty, vaihdettiin ohjelmointi Windows -ohjelmointiin, jossa toiminta tapahtuu Windows-ohjelmana. Seuraavana ominaisuutena mittautustieto siirrettiin tekstikenttään, josta tietoa on helppo leikata ja liittää muihin ohjelmistoihin sekä tallentaa tekstitiedostoksi.

Mittautulokset ovat edelleen alkuperäisessä wav-tiedoston muodossa, jossa tieto on näytejaksoissa. Tämän vuoksi tieto ei edelleenkaan ole vielä käsiteltä-

vissä suoraan kuin muutamalla ammattityökälulla. Tässä tapauksessa mittauksia on neljässä kanavassa.

Seuraavaksi näytteet sijoitettiin ruudukkoon, johon otettiin kokonaiset näytejonot sekä yksittäiset kanavakohtaiset mittaustulokset. Tulokset tallennetaan csv-tiedostoon. Lisäksi taulukko voidaan avata suoraan Excel-ohjelmistossa.

Mittatuloksia tutkittiin Excel-taulukoissa, jossa havaittiin, että ne eivät vastaa Audacity-ohjelman tietoja. Havaittiin, että numeerinen muoto ei ole oikea, vaan mittatuloksille pitää tehdä tavujärjestysmuunnos ennen muita käsittelyjä. Muutosten jälkeen työkaluun lisättiin vielä kaaviokuvat mittauksille (kuva 7).



KUVA 7. Käyttöliittymäkuva

Työkalu toimii yrityksen sisäisenä välineenä, jolla voidaan tutkia mittaustuloksia ja nähdä nopealla silmäyksellä, ovatko tulokset onnistuneita. Mittaustietoa käsit-

televää koodia on tehty eri vaiheissa monella tavoin, ja nämä ohjelmisto-osiot soveltuvat malleiksi tuleville sovelluksille.

Ohjelmisto ei sovellu suoraan BT-kokoonpanoissa käytettäväksi, mutta mittaus-tulosten käsitteleviä koodeja voidaan käyttää pienin muutoksin ja päinvastai- sessa järjestyksessä. Tällä tavoin saadaan tietokantaan yhteensopivaa tietoa laitteistosta riippumatta.

5 PÄÄTELMÄT JA JATKOKEHITTÄMINEN

5.1 Prosessit ja toiminta

Pienissä yrityksissä ei useinkaan ole vielä vakiintuneita tuotanto- tai johtamisprosesseja. Yrityskulttuurin luominen on alkuvaiheessa ja liiketoiminnalliset tavoitteet ovat avoimia ja joskus jopa muuttuvia.

Yrityksen alkuvaiheessa järjestelmäsuunnittelu on hyvin lähellä liiketoimintaa ja ne ohjaavat toisiansa. Isomman järjestelmän lähtötietokartoitus on haastavaa, koska osa asioista on vielä idean tasolla.

Lean-mallissa pyritään tehokkuuteen. Mallissa toteutetaan pienin mahdollinen toimiva tuote. Isossa järjestelmässä näitä tuotteita on useita. Minimituotteen suunnittelussa pitää kuitenkin huomioida tekemisen tehokkuus, jotta tuote palvelee myöhemmin järjestelmätason toiminnallisuuksia.

Alkuperäisenä ajatuksena oli tehdä pelidemo. Sellaisenaan toteutettu pelidemo ei olisi palvellut riittävästi järjestelmätasolla, koska se ei kytkeytyisi nykyiseen järjestelmään teknologisesti.

Yleisesti ohjelmistokehityksessä suositellaan standardoituja toimintatapoja, mutta nämä ovat yleisesti liian raskaita malleja pienyrityksiin. Ohjelmistotekniikka koostuu kuitenkin työkaluista, tavoista, prosessista ja laadusta. Pienikin yritys käyttää näitä, vaikka taso ei olisikaan standardin mukainen.

On väitetty, että standardin mukaisia menetelmiä ei tarvita, koska toimivan ohjelmistotuotteen voi toteuttaa ilman niitä, kevyesti tuottaen. Laadullisesti tällainen toiminta on riskinottoa, ja mitä monimutkaisempi kokonaisuus on, sitä suurempi riskikin on.

Laajan järjestelmän hallintaan prosessi pitää sopeuttaa yrityksen kokoon sekä pitää prosessihallinnan työmäärä mahdollisimman pienenä. Prosessin suunnit-

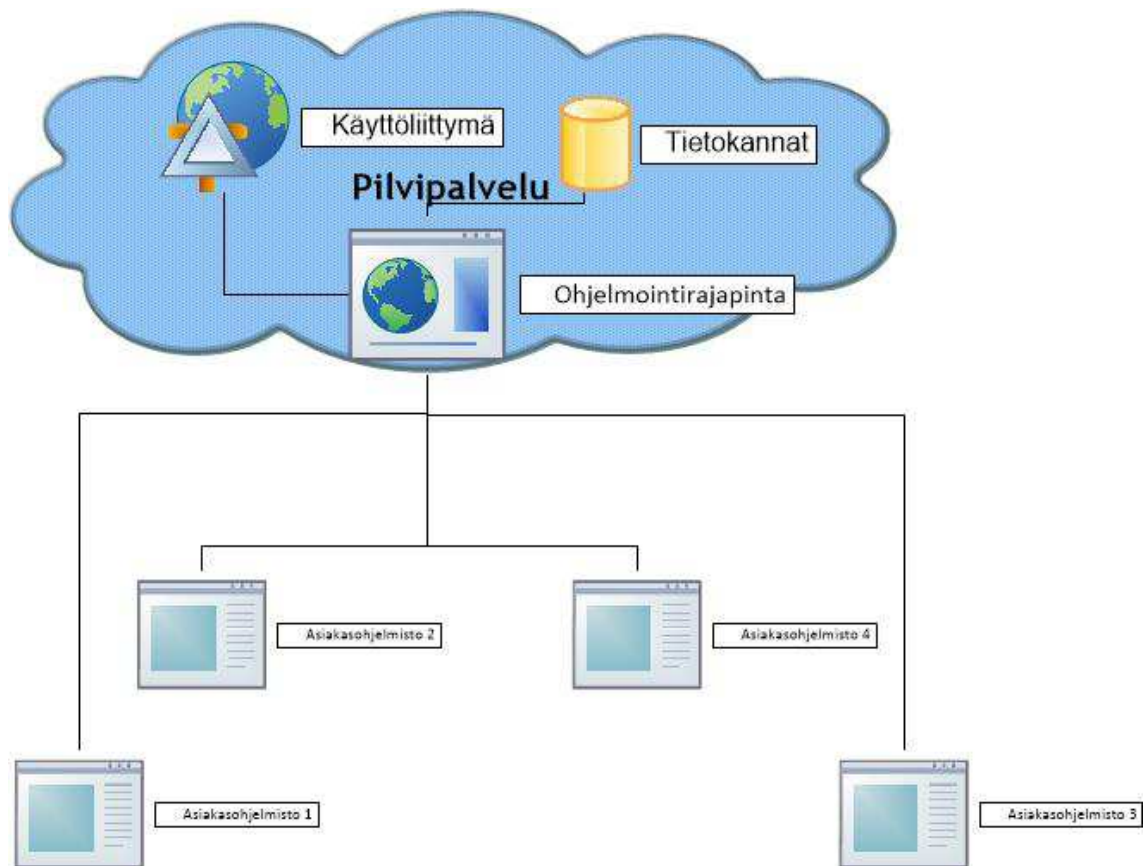
telussa on hyvä käyttää tavoitteena prosessin kypsyystasoja, jolloin hallinnointi on tavoitteellista ja se palvelee työn tekemistä.

5.2 Ohjelmistojen kokoonpano ja hallinta

Järjestelmässä olevia sisäisiä ohjelmistoja kannattaa samankaltaistaa sekä rajata erityyppisiä toimijoita muutamaan toimijaan, jolloin samoilla työkaluilla on helpompi tuottaa yhteensopivaa ohjelmistoa. Ohjelmistoja suunniteltaessa on kuitenkin hyvä muistaa lopputuotteen yhteensopivuus erilaisiin käyttäjillä oleviin laitteistoihin ja alustoihin. Lopputuotteen ensisijaisiksi laitteistoiksi ja alustoiksi kannattaa valita järjestelmää mukailevaa teknologiaa. Samankaltaisten teknologioiden vaatimustenmäärittäminen ja hallinta on helpompaa sekä projektin kokonaishallinta helpottuu.

Ongelmanratkaisumenetelmänä oli lineaarinen sekvenssimalli. Mallissa ongelma pilkotaan riittävän pieniin palasiin. Työkalun tekovaiheessa tuli vastaan useita vaikuttavia tekijöitä, jotka ovat järjestelmän sisäisiä vaatimuksia. Tällainen on esimerkiksi tietokannan ja ohjelmistojen väliin tarvittava ohjelmointirajapinta.

Tietokantaan ei kytkeydytä suoraan, koska tulevissa ohjelmistoissa käsitellään mahdollisesti asiakastietoja. Ohjelmointirajapinnan määrittely tulee ohjelmistojen toiminnallisista määreistä. Työkalun alkuvaiheessa mittaustuloksia syötettiin ensimmäisestä mittauksesta alkaen. Tällaisen määrän käsittely ei ollut järkevää, joten sitä rajoitettiin. Ohjelmointirajapinnan määrittely kannattaa aloittaa, kun BT-kokoonpanon tiedonlähetykseen tehtävää ohjelmistomodulia aletaan tehdä. Rajapinta voi käsitellä kaikkien asiakasohjelmistojen viestinnän, tai rajapintoja voi olla useampia, jos asiakasohjelmien tiedonkäsittely poikkeaa erilaisen toteutuksen vuoksi (kuva 8).



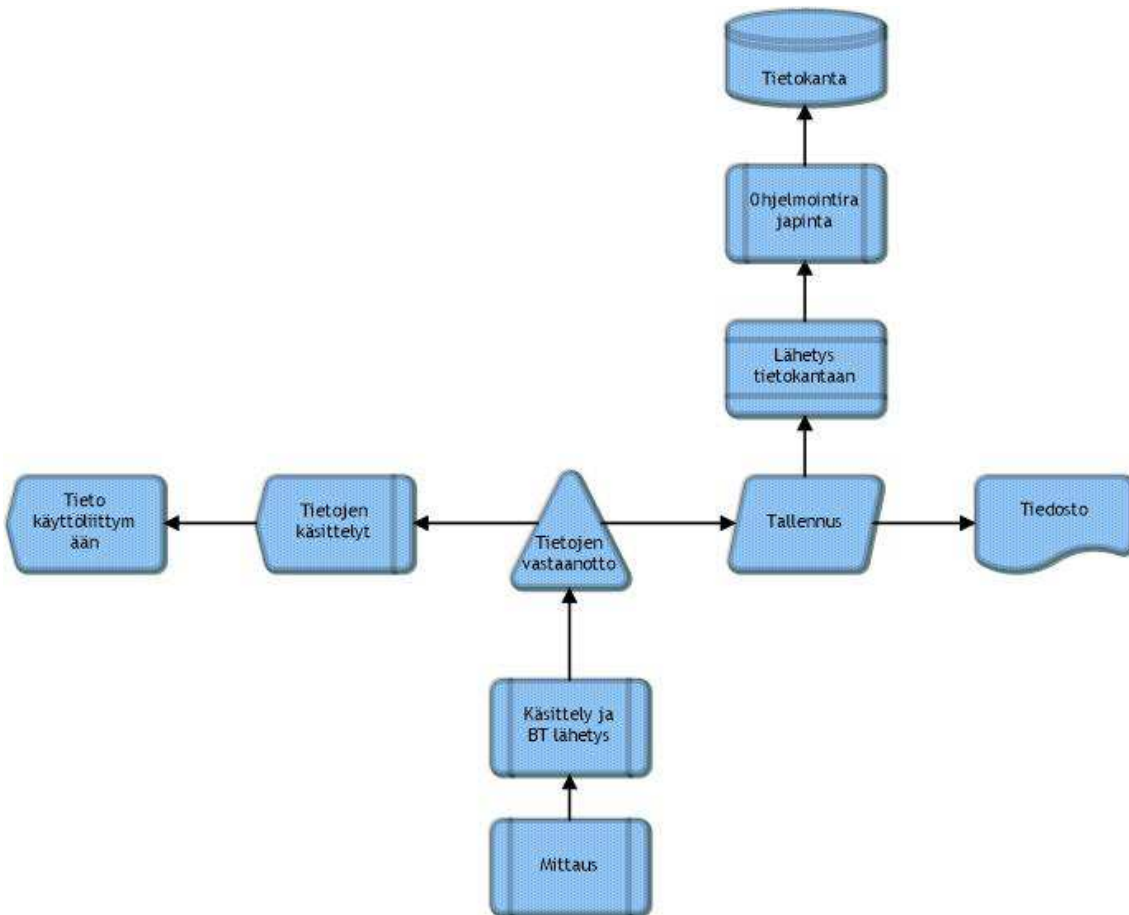
KUVA 8. Ohjelmointirajapinta

Ohjelmointirajapinnoilla voidaan hallita erityyppisten teknologisten ratkaisujen yhteensopivuutta. Tietoliikenne tietokantoihin hoidetaan yhtenäisellä tavalla, jolloin asiakasohjelmissa voidaan käyttää samaa ohjelmointia tai rakennetta tietoliikenteen hoitamiseen eikä asiakasohjelman toiminnallisuus vaikuta tähän.

Jotta ohjelmistorajapinta ei laajene liikaa, mittalaitteiston keräämä tieto on hyvä yhtenäistää eri teknologioiden välillä. Käytettävä kokoonpano ei saisi vaikuttaa, tietokantaan lähetettävään tietoon. Tiedon pitäisi olla samassa muodossa, jolloin tietokannan taulumäärä vähenee. Tietoa jäsentävän moduuli on hyvä toteuttaa ohjelmointirajapinnan yhtenä määrittävänä vaatimuksena.

5.3 Pelillisuus ja harjoitteet

Peliharjoitetta suunnitellessa on huomioitava tiedon varastointi asiakasohjelmassa ja tiedon lähettäminen tietokantaan (kuva 9). Tallennusmahdollisuus ja tiedonsiirtomuoto ovat teknisiä vaatimuksia asiakasohjelmiston laitteistolle ja järjestelmälle, jotka pitää arvioida mahdollisimman aikaisessa vaiheessa tapauskohtaisesti. Tekniset vaatimukset ovat pelisuunnittelussa rajaavia tekijöitä.



KUVA 9. BT-kokoonpanon toiminnallinen malli.

Peliteollisuudessa ei ole vielä vakiintunutta standardointia pelien tuottamiselle. Yleisesti pelien tekninen kartoitus rajoittuu järjestelmiin ja laitteisiin.

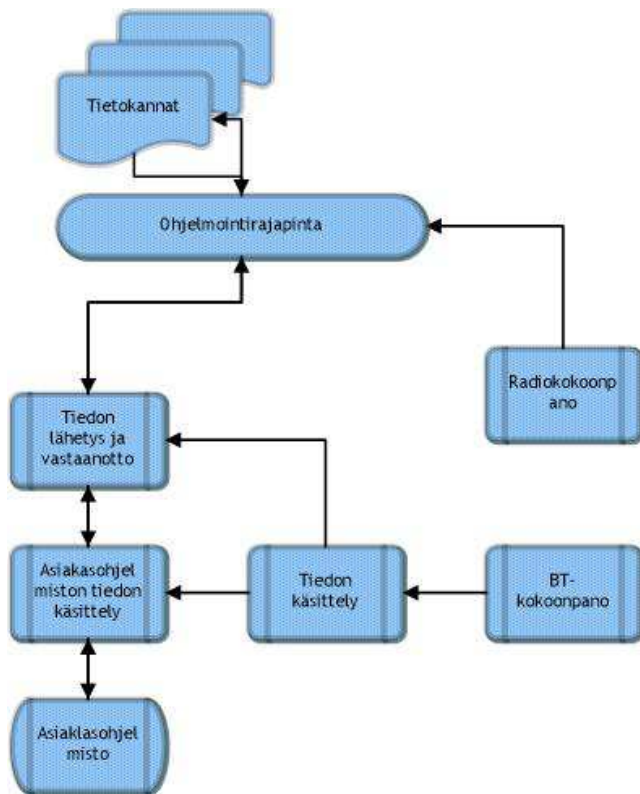
Peliharjoitteen tuottamiseksi järjestelmästä pitää olla riittävän tarkka määrittely teknisistä vaatimuksista. Pelidemo voidaan toteuttaa aikaisemmassakin vai-

heessa esittämään järjestelmän toimivuutta visuaalisesti. Toiminnallisen harjoitteen toteuttamiseksi ohjelmointirajapinta pitäisi olla määritetty ja toteutettu.

Järjestelmän yleisiä vaatimuksia ovat asiakastunnistetieto ja mittauslaitteiston tunnistetieto. Nämä muuttujat ovat yleisiä. Muuttujia käytetään normaaleissa ja pelillisissä harjoitteissa. Nämä tiedot ovat vakiomuotoista, ja niiden käsittely voidaan toteuttaa ohjelmassa omana osiona. Tällä tavoin saadaan ohjelmiston prototyyppi, jota voidaan käyttää useammassa erilaisessa sovelluksessa. Prototyyppi ja tietokannan ohjelmistorajapinta käyttävät samoja muuttujia ja vaatimuksia.

Mittalaitteen tietokannassa olevasta tunnistetiedosta pitää pystyä selvittämään mittalaitteisto, jolloin ohjelmallisesti pystytään määrittämään, millainen tiedon käsittely on. Koska mittalaiterakenteet ovat toiminnallisesti erilaisia, voidaan tällä tiedolla tarkistaa kyseisen kokoonpanon soveltuvuus asiakasohjelmaan sekä valita soveltuva ohjelmaosio käsittelemään tietoja.

Asiakasohjelmistolle tuleva mittatieto on syytä vakiinnuttaa mittalaittekokoonpanon toiminnallisuuden vaikuttamatta siihen. Radiokokoonpanossa tieto kulkee tietokannan kautta asiakasohjelmaan ja BT-kokoonpanossa se käsitellään suoraan asiakasohjelman laitteistossa. Asiakaslaitteistossa käsiteltävä tieto muotoillaan myös yhteensopivaksi tietokannan ohjelmointirajapinnan kanssa (kuva 10).

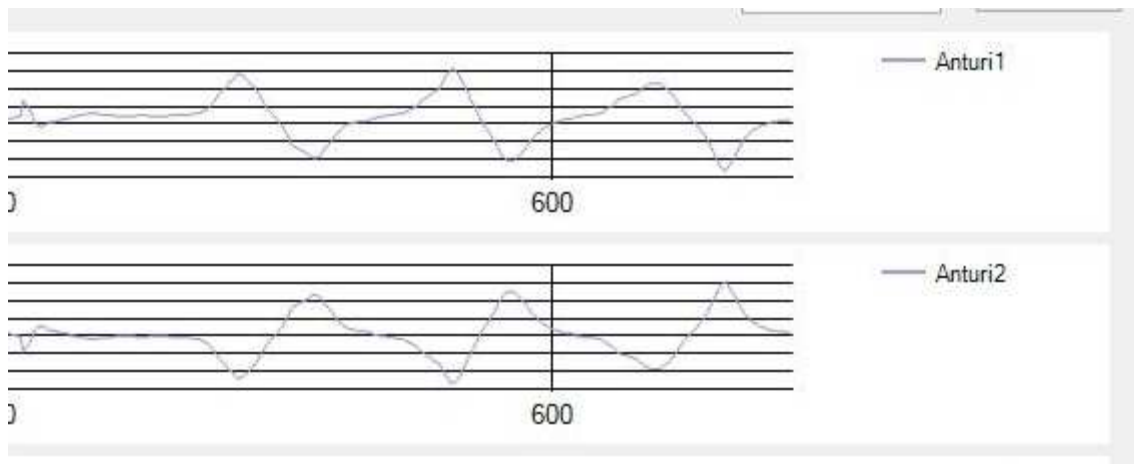


KUVA 10. Tietojen käsittelyt

Mittalaitetekoonpanojen tietoa voidaan käyttää ohjelmiston ohjaamiseen ja yleisiin visuaalisiin esitysmuotoihin, kuten taulukoihin, kaavioihin ja muuhun sellaiseen. Ohjaamiseen käytettävä tieto on syytä käsitellä mahdollisimman suoraan sellaisenaan, jotta asiakaslaitteisto ei kuormitu ja toiminnallinen viive saadaan mahdollisimman pieneksi. Mittatieto käsitellään näytejonona, josta poimitaan määritettyjen arvojen perusteella liipaisu-arvot. Nämä huomioidaan tiedon pakkaamisessa, joka määritetään mittalaitetekoonpanojen suunnitteluun.

Visuaalisesti muotoiltavien näytejonojen käsittelyyn vaikuttavat esitettävä määrä, esitysmuoto, käyttötarkoitus ja näytejonojen koko. Esitettävä määrä puuke kuvan visuaaliseen muotoon, mikä helpottaa käyttäjän informaation sisäistämistä. Tehdyssä työkalussa kaaviot muotoutuivat suoraan numeerisesta tiedosta, jossa tehtiin vain bittijärjestysmuunnoksia. Käyttötarkoituksena on tulkita mah-

dollisimman puhdasta mittaustulosta (kuva 11).



KUVA 11. Mittaustuloskaavio.

Asiakasohjelmaan tulevien visuaalisten esitysten tiedon käsittelyyn vaikuttaa myös reaaliaikaisuus. Tarvitseeko käyttäjä tiedon heti vai antaako tieto palautteen tehdystä jälkikäteen? Mittaustietoa on paljon ja sen käsittely kuormittaa laitteistoa. Mittalaitteiston ohjaimena käyttö on kriittinen tekijä toiminnallisuudelle, minkä vuoksi peliharjoitteissa kannattaa muu mittatulosten käsittely priorisoida pienemmäksi. Jälkikäteen tulostettavien visuaalisten esitysten tieto voidaan käsitellä suoraan tietokannassa. Tässä vaihtoehdossa on huomioitava tietoliikenteen määrä ja sen vaikutus asiakaslaitteen ja tietokannan välillä.

Pelillinen harjoite itsessään voi tuottaa järjestelmätason vaatimuksia. Pelissä syntyviä yksilöllisiä tuloksia voidaan käyttää kilpailun kasvattamiseen ja suorituksen toistettavuuteen.

Suoritteiden yleisen tason arviointia voidaan tehdä suoraan mittaustuloksista, mutta ohjelmistoissa voi olla myös jokin pisteytys pelikohtaisesti, josta saadaan yleinen sijoituslista muihin nähden. Tällaista tarvetta arvioidessa kannattaa huomioida, onko asiakasohjelma tehty kuntoutukseen vai yleiseen kuntoiluun.

Kuntoiluun suunnatuissa asiakasohjelmissa sijoituslista ja erilaisten sosiaalisten medioiden käyttö on huomioitava järjestelmässä. Sijoituslistatietoa suunnitellussa kannattaa huomioida yhteensopivuus eri harjoitteiden välillä, jolloin koko-

naisharjoittelua voidaan myös verrata sijoituslistamuodossa. Tällä tavoin eri harjoitteista voidaan tehdä tavoitteellisia kuntoilukokonaisuuksia kisailun muodossa. Järjestelmään voidaan myös toteuttaa sijoituslistan tietoihin perustuvaa palkitsemisjärjestelmää. Tällöin listatietojen täytyy olla yhtenäisiä eri harjoitteiden välillä. Harjoitekokonaisuuksiin ja yksittäisiin harjoitteisiin voidaan määrittää raja-arvot, joiden ylittyessä käyttäjä savuttaa määrätyn palkkion.

5.4 Järjestelmän hallinta

Järjestelmän teknologioiden vahvistuessa on syytä tehdä pääarkkitehtuurikuva, jossa näkyy eri teknologioiden tai ohjelmistojen vuorovaikutus. Ohjelmistot käyttävät usean eri teknologian toimintoja, joiden toiminnallisuus olisi hyvä koostaa vaatimusdokumentaatioon. Tällä tavoin saadaan alustaviin selvityksiin tarvittavaa järjestelmätietoa ja voidaan keskittyä kehitettävän ohjelmiston asiakasvaatimuksiin ja toiminnallisuuteen.

Pelillisen harjoittelun suunnitteluun järjestelmän vaatimukset antavat omat rajoitteensa, mutta laitteistotason rajoitteet ja vaatimukset pitää arvioida tapauskohtaisesti. Prosessien ja toimintojen tavoitteen määrittäminen seuraavaan kypsyystasoon antaa tavoitteellisen kehitysmallin päivittäiseen työhön ja järjestelmän rakentamiseen.

6 LOPPUSANAT

Työn päätavoitteena oli selvittää pelillisen harjoitteen toteuttamiseen liittyvät tarpeet järjestelmästä ja saada mahdollisimman selkeä kuva järjestelmän tarpeisiin pelillisten harjoitteiden toteuttamista varten.

Isoimpana haasteena ja ongelmana koin pelillisyyden toteuttamisen järjestelmässä, jonka osa-alueet koostuvat erilaisista sulautetuista järjestelmistä ja ohjelmistoista. Yleisesti tietokonepelit toteutetaan vakiintuneisiin ohjelmistoihin ja laitteistoihin, joissa on vakiintuneet rajapinnat eri toiminnoille. Pelialalla ei ole vakiintunutta tapaa tai menetelmää pelin sovittamiseksi erilaisiin järjestelmäintegraatioihin, mikä on iso haaste, varsinkin terveyspelien tuottamisessa.

Toisena tavoitteena oli ohjelmoida visuaalinen esitysmuoto pelillisiin elementein, mutta tämä ei ollut järkevää, koska työni edetessä huomattiin järjestelmän vaikuttavuuksia peliin. Tekniset vaatimukset eivät täytyneet vielä riittävässä määrin. Sen sijaan ohjelmointityössä toteutettiin järjestelmästä saatavien mittaustulosten käsittelyä helpottava työkalu.

Työn tekeminen antoi hyvän näkemyksen terveyspelien tuottamiseen liittyvistä ongelmista järjestelmien ja ohjelmistojen suhteen. Sain paljon hyvää kokemusta järjestelmän suunnitteluun ja ohjelmointityöhön. Työ oli sopivan haastava ja erittäin mielenkiintoinen.

LÄHTEET

1. Kaleva Jari-Pekka – Hiltunen Koopee –Latva Suvi 2013 Mapping the full potential of the emerging health game markets. Saatavissa. <http://www.sitra.fi/julkaisut/Selvityksi%C3%A4-sarja/Selvityksia72.pdf> Hakupäivä 30.10.2013
2. BonWell Saatavissa <http://www.bonwellin.com/fi/> Hakupäivä 30.10.2013
3. Tracy Diziere – Associates 2010 Process Development For Small Business 040610 Tda. Saatavissa. <http://www.slideshare.net/tracydiziere/process-development-for-small-business-040610-tda> Hakupäivä 18.11.2013
4. Ryan Tim The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal. Saatavissa. http://www.gamasutra.com/view/feature/131791/the_anatomy_of_a_design_document_.php Hakupäivä 12.1.2013
5. Pressman Roger – Ince Darrel 2000. Software engineering. A practitioner's approach. European adaptation. Lontoo: McCraw-Hill International Ltd.