



Ville Sarkki

Toiminnanohjausjärjestelmän palvelinasennuksen määrittäminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tuotantotalouden tutkinto-ohjelma

Insinööriyö

2.6.2023

Tiivistelmä

Tekijä:	Ville Sarkki
Otsikko:	Toiminnanohjausjärjestelmän palvelinasennuksen määrittäminen
Sivumäärä:	79 sivua + 1 liite
Aika:	2.6.2023
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tuotantotalouden tutkinto-ohjelma
Ammatillinen pääaine:	Kansainvälinen ICT-liiketoiminta
Ohjaajat:	Yliopettaja Antero Putkiranta Osaamisaluepäällikkö Janne Salonen

Insinööriyön päätavoitteena oli hahmotella organisaation sisäisesti Metropolia Ammattikorkeakoululle avoimen lähdekoodin toiminnanohjausjärjestelmän, webERP:n, asentamisen prosessi sekä luoda sen toteutukseen liittyvät ohjeet.

Työssä arvioitiin aluksi, mikä olisi resursseiltaan oikeinmitoitettu palvelinpalvelu, joka tilattaisiin organisaation tietohallinnon palvelukatalogista. Lisäksi pyrittiin muodostamaan näkemyksiä palvelimen käyttämisen tietoturvaan liittyviin seikkoihin.

Lisähaasteena, ja oppimiskokemuksena, lähdettiin samalla kehittämään työkalua asennusprosessin automatisointia varten, helpottamaan hallinnoijan työtä entisestään. Kokonaisvaltaisen ja hienostuneen ratkaisun kehittäminen olisi ollut laajuudessaan liiallista, varsinkin suhteessa hyötyihin, joten työssä keskityttiin vain selkeiden ohjeiden laatimiseen palvelimen asentajaa varten.

Insinööriyön saatossa pyrittiin löytämään ratkaisuja omien toimintatapojen parantamiseen projektinhallinnallisten teemojen ympäriltä. Kartoitustyön vaikutukset olivat jokseenkin subjektiivisia opeiltaan, mutta niitä dokumentoitiin reflektoinnin nimissä työhön.

Päätavoite täytettiin dokumentoimalla asennusprosessiin liittyvät ohjeet hallinnoijaa varten palvelimen tilaamisen ja siihen webERP:n asentamisen osalta.

Avainsanat: Toiminnanohjausjärjestelmä, webERP

Abstract

Author: Ville Sarkki
Title: Defining the process for server installation of an ERP system
Number of Pages: 79 pages + 1 appendix
Date: 2 June 2023

Degree: Bachelor of Engineering
Degree Programme: Industrial Engineering and Management
Professional Major: International ICT Business
Supervisors: Senior Lecturer Antero Putkiranta
Head of School, ICT Janne Salonen

The purpose of this thesis was to determine the process of installing an open-source ERP system, webERP, into a server and provide documented instructions on how this can be accomplished. The task was commissioned by Metropolia University of Applied Sciences.

In the beginning stages of the thesis, sizing options of the server were considered as the server would be ordered for use from Metropolia's service catalogue. Views on cyber security were also contemplated.

For an added extra objective, which was conducted simultaneously with the main objective, constructing an automated installation tool was initially attempted but this turned out to be a too extensive goal to be pursued, especially considering the small added benefit of it and the focus was shifted toward producing workable instructions for the installation process.

While working on the thesis, ways to improve one's own working habits were sought after by reading theory relating to agile practices and methods. The conclusions are subjective in nature, but they were documented for reasons of reflecting on the work that was done.

The main objective was satisfied by documenting a workable procedure of the installation process regarding ordering the server and getting webERP running on it.

Keywords: Enterprise Resource Planning, webERP

Sisällys

1	Johdanto	1
2	Tavoite ja työprosessi	2
2.1	Työn tavoite	2
2.2	Työn rajaus	2
2.3	Työn rakenne	3
3	Tietojärjestelmän asentamisesta	4
3.1	Toiminnanohjausjärjestelmä	4
3.1.1	Historia	4
3.1.2	Näkemyksiä määritelmästä	5
3.1.3	Järjestelmän käyttöönottoprosessi	7
3.1.4	Haasteita käyttöönotossa	10
3.2	webERP	12
3.3	Tietojärjestelmäkäsitteitä	16
3.4	Palvelimen resurssitarpeet	20
3.5	Näkökulmia tietoturvaan	23
3.6	Projektinhallinnallisia seikkoja	26
3.6.1	Projekti	26
3.6.2	Ketterät menetelmät	29
4	Toteutus	37
4.1	Työmenetelmien tarkastelu	37
4.2	Ketterät menetelmät	38
4.3	Näkemyksiä palvelimen valintaan	41
4.3.1	webERP:n vaatimusten täyttäminen	41
4.3.2	Käyttöjärjestelmän käyttökokemus	43
4.3.3	Järjestelmän tehokkuus	44
4.3.4	Elinkaarisuunnittelu	44
4.4	Asennusohjeet	45
4.4.1	Palvelimen tilaaminen	47
4.4.2	Käyttäjätilit	59
4.4.3	webERP:n asentaminen palvelimelle	63
4.5	Asennustyökalu	73
5	Yhteenveto	75

1 Johdanto

Ajan saatossa ja tekniikan kehittyessä tietojärjestelmien hinnat ovat pudonneet. Yrityksillä, joilla on pienemmät resurssit, on mahdollisuus hankkia itselleen toiminnanohjausjärjestelmä, joita on nykyään tarjolla jopa ilmaisversioina ja avoimella lähdekoodilla.

Insinööriyön ensisijaisena tarkoituksena oli muodostaa ohjeet webERP-toiminnanohjausjärjestelmän asennuksesta. Ohjeet muodostaisivat vakioituneen ja toistettavan prosessin, jonka mukaan asennuksen voisi toteuttaa. Työ tehtiin Metropolia Ammattikorkeakoulun opetuskäyttöä varten. Kantaa otettiin myös tietoturva-asioihin.

Aluksi kartoitettiin palvelukatalogista tilattavan palvelimen tarvevaatimusta. Arvion muodostamiseen käytettiin erilaisia lähteitä, mutta syvällisemmän ja varmemman käsityksen saavuttamiseksi olisi vaadittu joko testaamista tai webERP-järjestelmän käytön kokemuspohjaa.

Asennusalustan määrittelyn jälkeen tutustuttiin erilaisiin tietojärjestelmiin ja ohjelmiin, minkä lomassa tehtiin alustava toimiva versio asennusohjeista. Sittemmin ohjeita paranneltiin ja siitä siivottiin turhaksi havaittuja osia pois.

Työssä päädyttiin asennusprosessin tehostamiseksi sekä oppimiskokemuksena, kokeilemaan asennustyökalun rakentamista. Tässä tavoitteena oli muodostaa mahdollisimman pitkälle automatisoitu työkalu asennusta ja kenties hallinnointiakin varten. Tämä sivutavoite oli liian laaja eikä kokonaisvaltaisesti toimivaa työkalua muodostettu.

Insinööriyössä opiskeltiin toiminnanohjausjärjestelmien historiaa ja teoriaa, minkä lisäksi teoriapohjaa kartoitettiin tietojärjestelmien käyttöönottoprosesseihin liittyen. WebERP:n asennusohjeet Linux-palvelimelle ovat vain pieni osa kokonaisvaltaista tietojärjestelmän käyttöönottoprosessia.

Työnteon ja itsensä johtamisen tueksi päätettiin omatoimisesti tutkia ketteriä menetelmiä ja pohtia, olisiko sieltä saanut sovellettua joitain käytänteitä oman työn tueksi. Päätös perustui oppimiskokemuksen muodostamiseen työtehon ja -motivaation tukemisen ohella, mutta lisäopiskelu vaikutti negatiivisesti itse työn läpimenoaikaan.

2 Tavoite ja työprosessi

Tässä luvussa kuvataan, miten insinööriytyö eteni. Tarkoitus on muodostaa yleiskuva sen tavoitteista, kronologisesta etenemisestä sekä käydä läpi työvaiheet ja työvaiheiden lopputuloksia.

2.1 Työn tavoite

Insinööriytyön päätavoite oli laatia ohjeet webERP-toiminnanohjausjärjestelmän asentamisesta käyttövalmiuteen. Myös tietoturvaan liittyviä asioita pohdittiin. Ylimääräisenä lisätavoitteena kehitettiin skriptipohjaista asennustyökalua, jonka laajuus oppimisprojektina olisi muodostunut lopullisen ja mahdollisimman pitkälle automatisoidun version kannalta turhan laajaksi.

Lisätavoitteen suorittamisen aikana opiskeltiin näkökulmia omia työmenetelmiä tukeviin seikkoihin, jolloin päädyttiin opiskelemaan ketteriä menetelmiä ja etsimään soveltuvuutta koodin rakentamiseen. Koettiin, että opitusta voidaan ammentaa uudenlaista hahmotuskykyä myös itse insinööriytyön kirjoittamista kohtaan.

2.2 Työn rajaus

Asennusohjeet koskevat Linux-palvelinta. Windowsia, tai muita järjestelmiä, varten ei luotu ohjeita.

Kantaa ei otettu siihen, miten webERP:tä tulisi käyttää; se haluttiin ohjeiden kautta vain käyttövalmiiksi, joskin asennusohjeiden loppuun annettiin pikaohje käyttäjien lisäämisestä järjestelmään.

Palvelimen toimintaan ei paneuduttu ylimääräisesti. WebERP asettaa tiettyjä vaatimuksia palvelinta kohtaan, mutta niiden täyttämisen jälkeen ei lopputulosta jääty parantelemaan. Palvelimen hallintaan yleisesti ei oteta kantaa, mutta asennusohjeissa luodaan hallinnointia varten käyttäjätilit.

2.3 Työn rakenne

Insinööryö on jaettu teorian, ja sitä seuraaviin käytännön osioihin: lukuihin 3 ja 4. Teoriaosuudessa luvussa 3 käsitellään toiminnanohjausjärjestelmiä ja niiden käyttöönottoprosessia, tietoteknisiä käsitteitä ja muodostetaan näkökulmia projektinhallinnallisiin seikkoihin, joiden parissa oli tarkoitus löytää ratkaisuja parempaan itsensä johtamiseen.

Toteutuksen eli ratkaisuehdotuksen osuuden luvun 4 alussa käydään läpi projektin läpimenoon liittyviä opittuja asioita alaluvuissa 4.1 ja 4.2. Itse tietotekninen ratkaisu alkaa resursseiltaan sopivan palvelinpaketin pohtimisella alaluvussa 4.3. Viitteitä haetaan webERP:n omista materiaaleista, muista ERP-järjestelmistä ja yleisistä arvioista palvelimien kokoon ja suorituskykyyn liittyen.

Päätavoitteena ollut palvelimen asennus ja sen ohjeet alaluvussa 4.4 on jaettu edelleen omiin alalukuihinsa: palvelimen tilaamiseen Metropolian Educloud-palvelusta alalukuun 4.4.1, käyttäjätilien muodostamiseen hallinnoijaa varten alalukuun 4.4.2 ja itse varsinaisiin asennusohjeisiin alalukuun 4.4.3. Asennusprosessi esitellään taulukkomuodossa alaluvun 4.4 taulukossa 1.

Palvelimien hallinnoinnissa yleinen paras käytäntö on välttää juurikäyttäjätilien käyttämistä. Hallinnollisia asioita harjoiteltiin ja CentOS-käyttöjärjestelmään tutustuttiin ennen kuin insinööryöhön saatiin sisältöä. Käyttäjätilien osalta dokumentoitiin lyhyesti hallinnollisten tilien luominen CentOS- ja MariaDB-järjestelmiin.

Alaluvussa 4.5 käydään läpi asennustyökalun laatimista insinööriyön aikana. Kehitystyö kasvatti liikaa projektin tavoitteita, joten sen laadintaa ei jatkettu toimitettavaan muotoon asti.

Työstä esitetään vielä yhteenveto luvussa 5. Siinä kerrataan ja pohditaan tehtyä työtä sekä sen tuloksia.

3 Tietojärjestelmän asentamisesta

Tässä luvussa käsitellään teoriapainotteisesti insinööriyöhön liittyviä käsitteitä ja haasteita. Tietojärjestelmiin liittyvän teorian lisäksi insinööriyön aikana tarkasteltiin projektinhallinnallisia seikkoja, joiden tavoitteena oli löytää tukevia omien työtapojen tehostamiseen. Teoriaan nojautuen esitetään seuraavassa luvussa 4 pohdintoja ja ratkaisuja, eritoten insinööriyön päätavoitteena luodut webERP-toiminnanohjausjärjestelmän asennusohjeet.

3.1 Toiminnanohjausjärjestelmä

Tässä alaluvussa käydään läpi toiminnanohjausjärjestelmää (Enterprise Resource Planning, lyhennetään ERP) käsitteenä sekä sen käyttöönottamista prosessina, joka voi osoittautua laajaksi ja yrityskontekstissa vaikeasti hallittavaksi kokonaisuudeksi. Toiminnanohjausjärjestelmän käyttöönottoprosessi ei kuulu työn käytäntöosioon, mutta sitä päätettiin käsitellä sivistävästä syistä.

Seuraavaksi käsitellään termin historiaa ja pohditaan sen määritelmää. Sen jälkeen käsitellään tietojärjestelmän käyttöönottamista prosessina sekä siihen liittyviä haasteita, eritoten pk-yrityksen näkökulmasta.

3.1.1 Historia

Toiminnanohjausjärjestelmien kehityskaari nähdään yleisesti alkaneen 1960-luvulla varastonohjauksen parissa, josta seuraava virstanpylväs oli 1970-luvun MRP-järjestelmät (Materials Resource Planning; materiaalitarkvuunnittelu),

joissa materiaalitovelaskentojen avulla ohjattiin ostotoimintaa ja automatisoitiin tilausten tekemistä. (Kettunen & Simons 2001: 46.)

Järjestelmät olivat alkuun raskaita ja kalliita, joten niitä oli liiketoimessa käytössä vain isoilla yrityksillä. 1980-luvulla kehitettiin MRP II, joka toi lisää toiminnallisuutta MRP:hen. Ennen pitkää erilaisia osa-alueita, kuten projektinhallintaa ja taloushallintoa, alettiin nivoa yhden järjestelmän alaisuuteen ja monialaista tietojärjestelmäkonseptia alettiin nimittää 1990-luvulla ERP:ksi.

Vanhojen aikojen massiiviset konesalit voidaan nykyään korvata kotitoimistoon hankittavilla komponenteilla. Datan käsittelyn kehitys, sekä internet, loivat uuden aikakauden, ERP II:n, jonka nähdään suurin piirtein alkaneen 2000-luvulla.

Sittemmin ERP:tä on alettu tarjoamaan pilvipalveluna, jossa yritys on toiminnanohjausjärjestelmänsä yhteydessä etänä, tietokoneohjelman tai jopa nettiselaimen kautta. Yritys voi myös ulkoistetussa palvelussa jättää järjestelmän ylläpitämisen asiantuntijan vastuulle.

ERP-järjestelmien tulevaisuuden kehitysnäkymissä nähdään järjestelmien automaation ja integroinnin laajenemista koneoppimisen ja tekoälyn avustamina. Esimerkiksi IoT-trendi tuo enemmän laitteita mukaan järjestelmään, ja prosessien hallinnassa hyötykäytettävän datan määrä kasvaa.

3.1.2 Näkemyksiä määritelmästä

Kalliokoski ym. näkevät, että ”toiminnanohjauksella ohjataan yrityksen työtä ja resursseja. Työn tulosten tulee olla asiakkaan vaatimusten mukaisia ja valmistua asiakkaalle luvatussa ajassa” ja että ”resurssien [kuten henkilöstön ja laitteiden] tehokas käyttö muodostaa perustan taloudellisesti kannattavalle toiminnalle”. He jakavat yrityksen toiminnan ohjauksen kolmeen tasoon: strategiseen, kehitystoiminnalliseen ja operatiiviseen ohjaukseen. (Kettunen & Simons 2001: 41–42.)

Toiminnanohjausjärjestelmä on liiketoimintaprosessien hallintaohjelmisto, jonka avulla voidaan hallita yrityksen taloushallintoa, toimitusketjua, toimintoja, raportointia, valmistusta ja henkilöstöhallinnon toimintoja sekä integroida ne yhdeksi kokonaisuudeksi. (Microsoft: Mikä on ERP?)

IT-suuryritys Oraclen sivuilta saatavassa dokumentissa mainitaan, että ERP auttaa organisaation eri osastoja, ja niiden liiketoiminnallisia prosesseja, yhdistämään toimintaansa saman järjestelmän (ja datan) alle (Tittel 2022: 1–2).

Käsitteitä toiminnanohjaus ja toiminnanohjausjärjestelmä käytetään eri lähteissä hieman eri tavoin. Insinööriyössä puhutaan toiminnanohjausjärjestelmästä ERP:nä sen tietojärjestelmäasiayhteydessä.

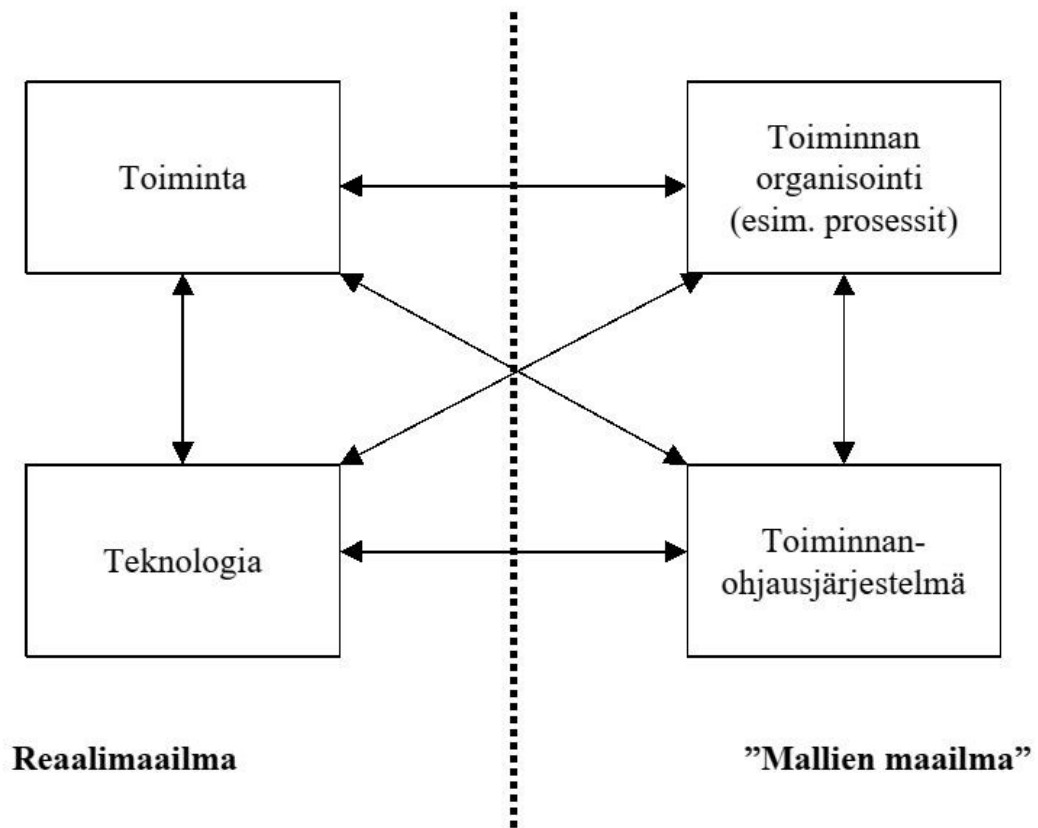
Lopullista määritelmää ERP:lle ei pyritty löytämään käsitteen monialaisuuden takia, sillä kuten Reijonen ym. ilmaisevat: ”yrityksen toiminnanohjaus ja sitä tukevat tietojärjestelmät ovat sekä strategisesti että operatiivisesti merkittäviä, sillä ne integroivat yrityksen ydinprosessit” (Kettunen & Simons 2001: 69), ja etenkin, kuten Nurminen ja Järvinen toteavat: ”toiminnanohjausjärjestelmät ovat tietojärjestelmiä, jotka koskettavat tyypillisesti organisaation kaikkia toimintoja” (Kettunen & Simons 2001: 169).

Toiminnanohjausjärjestelmän voisi siis tiivistää olevan (tieto)järjestelmä, joka pyrkii monialaisesti tukemaan yrityksen liiketoimintaa, halliten tietotekniikan avulla erinäisiä toiminnan alueita yhtenä kokonaisuutena.

Karvonen ja Tommila linjaavat käyttötarkoituksen osalta, että ”järjestelmän tulisi vahvistaa yrityksen strategian toteutumista ja auttaa seuraamaan ja ohjaamaan niitä tavoitteita ja tekijöitä, jotka ovat yrityksen kilpailukyvyn kannalta oleellisia” (Kettunen & Simons 2001: 132).

Toiminnanohjausjärjestelmät luodaan mallien pohjalta kuvaamaan tosielämän liiketoiminnan prosesseja; seikka, jonka Halttunen ym. nostavat esille kuvassa 1 esille (Kettunen & Simons 2001: 157–158). Näin fyysistä käytännön tasoa siirre-

tään dataksi, jota puolestaan koneet voivat prosessoida ja hallinnoida. Halttunen ym. täsmentävät, että sitä parempaa lopputulosta voidaan toivoa, mitä paremmin reaali maailma saadaan mallinnettua.



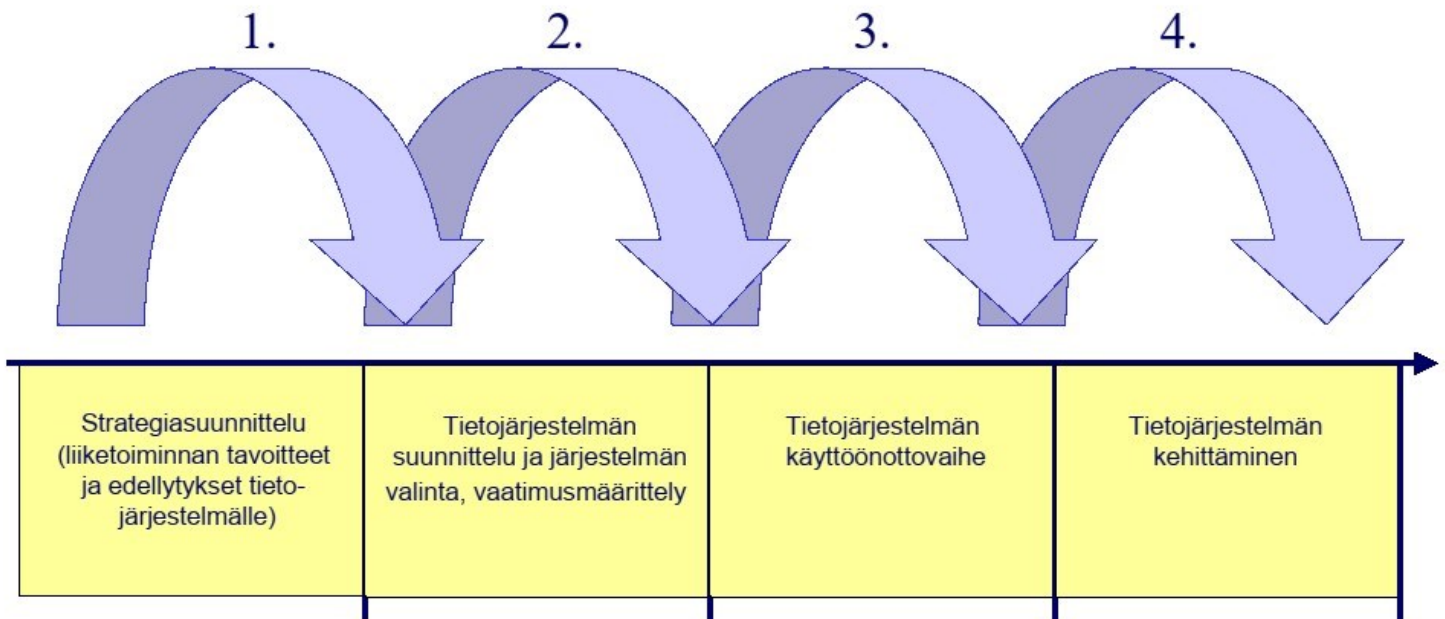
Kuva 1. Reaalimaailman toimintojen mallinnus on ERP:n ydinnäkökulma (kuvan lähde: Kettunen & Simons 2001: 158).

Malleja puolestaan hallinnoidaan tietojärjestelmien kehitystä varten luotujen menetelmien kautta. Menetelmiä on lukuisia.

3.1.3 Järjestelmän käyttöönottoprosessi

Hyötyläinen ja Kalliokoski (Kettunen & Simons 2001: 24) esittelevät kuvassa 2 nelivaiheisen mallin, joka koskee yleisesti tietojärjestelmien (kattotermi, johon toiminnanohjausjärjestelmä sisältyy) käyttöönottoprosessia, jota kuvailtiin koko organisaatiota koskeväksi oppimisprosessiksi (Kettunen & Simons 2001: 28–29;

31). Onnistumisen kannalta on tärkeä muistaa, että tietojärjestelmästä saatu hyöty realisoituu vasta käytössä, joten sen tulisi vastata oman toiminnan tarpeita.



Kuva 2. Tietojärjestelmän käyttöönottoprosessi Hyötyläisen ja Kalliokosken mukaan (kuvan lähde: Kettunen & Simons 2001: 24).

Strategiasuunnittelu

Lähtökohta on yrityksen liiketoiminnallisen strategian peilaaminen siihen, miten tietojärjestelmä voisi tukea sitä. Tietojärjestelmien elinkaaria ei yleensä suunnitella vain lyhytaikaiseen käyttöön ja omien tavoitteiden tulisi olla riittävän selkeät, mikä auttaisi sopivan järjestelmän hankkimisessa.

Tietojärjestelmän suunnittelu, järjestelmän valinta ja vaatimusmäärittely

Tässä vaiheessa tarkennetaan sitä, miten tietojärjestelmän odotetaan täsmällisesti toimivan. Yrityksen nykytilan analysointi, tai vähintään tuntemus, on tarpeen. Kaupallisten järjestelmien toimittajien kanssa voidaan käydä keskusteluita yrityksen tarpeeseen liittyen, jolloin nykytilan ja sen tarpeiden tunnistaminen olisi oleellista ja johtaa yrityksen kannalta parempaan asemaan myyntineuvotte- luissa.

Tietojärjestelmän käyttöönottovaihe

Kun järjestelmä on valittu, on seuraavaksi edessä liiketoiminnan siirtäminen tietojärjestelmään. Tähän vaiheeseen sisältyy ohjelmiston mahdollinen räätälöinti. Käyttöönotto vaatii organisaation sisältä tiivistä fokusta, mikäli se etenisi vai- voitta.

Tietojärjestelmän kehittäminen

Yrityksen tulisi huomioida kuinka tietojärjestelmä palvelee liiketoimintaa myös jatkossa. Iso haaste on esimerkiksi, jos toiminnan luonne vaatii muutoksia, jotka heijastuvat vaatimuksena toimittaa tietojärjestelmään ohjelmistopäivityksiä. Toi- saalta huoli saattaisi olla myös henkilöstön IT-osaamisen riittävydessä tulevai- suudessakin.

Tietojärjestelmänkin (kuten muiden yritystoiminnan alueiden osalta) kohdalla tu- lisi siis pyrkiä jatkuvaan kehittämiseen (Kettunen & Simons 2001: 24–25). Li- säksi Reijonen ym. toteavat, että ”yrityksen johdon keskeinen tehtävä on integ- roida tietojärjestelmien kehittäminen kiinteästi yrityksen liiketoiminnan kehittämi- seen, mikä on toteutettava aina strategiatasolta jokapäiväiseen operatiiviseen toimintaan” (Kettunen & Simons 2001: 70).

3.1.4 Haasteita käyttöönotossa

Kettunen ja Simons nostavat kolme keskeistä eroa, miten pk-yrityksen tietojärjestelmän käyttöönottoprosessi eroaa suuren yrityksen vastaavasta. Pk-yritysten toiminnan luonne ja toimintatavat ovat vähemmän muodollisia, resursseja on rajallisemmin käytössä ja osaamis pohja etenkin tietoteknisessä mielessä ei oletettavasti ole yhtä vankalla tasolla (Kettunen & Simons 2001: 221).

Eräs pk-yrityksen kilpailuvaltti verrattuna suureen yritykseen on toiminnan ketteryydessä – pienimuotoisempaa liiketoimintaa voidaan muuttaa nopeammin vastaamaan markkinoiden tarpeita. Olisi riskialtista hankkia raskas ja markkinoiden muutoksiin huonommin taipuva toiminnanohjausjärjestelmä, etenkin jos se olisi vielä taloudellisesti merkittävä investointi (pienemmän yrityksen resursseille). (Kettunen & Simons 2001: 220–222).

Resurssien vähyys ilmenee moninaisesti, eräs näkökulma rahan lisäksi on henkilöresurssit: onko työntekijöillä riittävä IT-osaaminen ja onko yrityksellä ylipäättään aikaa sitoutua käyttöönottoprosessiin, joka voi vaatia oman toiminnan kartoittamista, liiketoimintaa siirrettäessä tietojärjestelmiin.

Seuraavana tarkastellaan erilaisia näkökulmia ERP:n käyttöönottoon liittyen.

Jäykkyys

Reijonen ym. toteavat, että ”dynaamisessa ja ennustamattomassa toimintaympäristössä organisaatioiden muutokset eivät ole lopullisia ja täydellisiä. Näin ollen on orientoiduttava toiminnan jatkuvaan kehittämiseen” (Kettunen & Simons 2001: 70).

Toiminnanohjausjärjestelmän jäykkyys (sen muovaumattomuus liiketoiminnan ympärille) nähdään yleisenä haasteena. Lähestyminen järjestelmien luomisessa on ollut standardinomaisesti rakentaa järjestelmä palvelemaan tyypillisimpiä toimintaprosesseja, mikä tuottaa haasteen monipuolisesti toimivalle yritykselle, jolla olisi tarve muuntaa toimintaansa. Tämä on johtanut joskus liiketoiminnan

sovittamiseen järjestelmän ympärille, ikään kuin päinvastoin alkuperäiseen tarkoitukseen nähden. (Kettunen & Simons 2001: 49–50).

Pitkä käyttöönottoprosessi

Järjestelmän käyttöönottaminen saattaa muodostaa ajallisen haasteen. Vuosituhannen alussa nähtiin Kalliokosken ym. mukaan keskimääräisen käyttöönottamisen suunnitteluineen kestäneen noin vuoden (Kettunen & Simons 2001: 50). Riski on ilmeinen; liiketoiminta saattaa muuttua ja jäykkä ohjelmisto ei enää vastaakaan tarpeisiin tarpeeksi.

Tosielämän esimerkki on Lidlin vuonna 2011 alkanut ERP-projekti ”eLWIS”, johon investoitiin puoli miljardia euroa ja se kuopattiin vuonna 2018, jolloin ”palatettiin lähtöruutuun”.

Kehitys

Kalliokoski ym. mukaan toiminnanohjausjärjestelmien kehitys on ollut vauhdikasta ja sen kautta on avautunut väyliä liiketoiminnan tehostamiseen ja toisaalta uusien palvelukonseptien luomiseen (Kettunen & Simons 2001: 55–56). Reijonen ym. toteavat, että ”yritykset joutuvat kuitenkin jatkuvasti tehostamaan ja kehittämään toiminnanohjaustaan ja prosessejaan, mikä usein edellyttää myös tietojärjestelmien kehittämistä.” (Kettunen & Simons 2001: 69.)

Inhimillisenä näkökulmana mainittakoon, että liiketoimintaympäristön muutokset ja henkilöstön osaamisen vaatimukset nähtiin eräässä tutkimuksessa stressin ja uupumuksen aiheuttajina henkilöstöllä. Reijonen ym. näkevät, että yhtäältä työ on muuttunut tietojärjestelmien kautta yksitoikkoisemmaksi ja helpommaksi, mutta toisaalta toimihenkilöillä on haaste työn kasvavassa abstraktiudessa (Kettunen & Simons 2001: 67).

3.2 webERP

WebERP on PHP:llä tehty avoimen lähdekoodin toiminnanohjausjärjestelmä, jota alettiin kehittää 2000-luvun alussa. Se on web-sovellus, eli sitä käytetään käyttäjän valitseman internetselaimen kautta. Tämä luo mahdollisuuden käyttää webERP:tä missä vain, kunhan käytettävällä laitteella ja webERP:n palvelimella on internetyhteys kunnossa. Sen ohjekirjassa nostetaan kehitystyön kulmakinä avoimen lähdekoodin ja ilmaisohjelmiston lisäksi ohjelmiston keveyden mukanaan tuoma tehokkuus ja käytettävyys.

WebERP:tä voi kokeilla julkisesti internetin kautta sen kotisivujen kautta osoitteessa www.weberp.org/weberp/. Tämä malliohjelma ei ollut insinööriyön aikana aina saatavilla eikä siinä välttämättä ole saatavilla viimeisimpiä muutoksia, jotka löytyvät GitHubin kautta ylläpidetyssä viimeisimmässä versiossa. Tiedoksi, että webERP:n GitHub-sivu sijaitsee osoitteessa: github.com/timschofield/webERP-svn.

Vaatimukset

Toimiakseen webERP asennetaan palvelimelle, joka pystyy tulkitsemaan PHP-kieltä, josta löytyy nettipalvelin ja jolla on MySQL (tai MariaDB) tietokantajärjestelmänä.

Asiasta kiinnostunut henkilö voi melko vaivatta ladata ja asentaa itselleen kotikoneelleenkin jonkin esivalmistellun pinon. Kotikäytössä webERP:tä käytettiin aluksi omalta Windows-koneelta paikallasennuksena XAMPP-pinon avulla. Ohjelmiston voi halutessaan ladata osoitteesta: www.apachefriends.org/download.html. Se mainittiin myös webERP:n ohjekirjassa ja sen käyttö oli tehty melko helpoksi ja käyttäjäystävälliseksi graafisen käyttöliittymän avulla, kun verrataan ohjelmistojen asentamista ja käyttämistä pelkän tekstipohjaisen komentorivin kautta.

Muun muassa XAMPP:ssa ja LAMP:ssa tulee mukana webERP:n edellä mainitut vaaditut ohjelmistot. Näistä jälkimmäinen löytyy Educloudin katalogista.

Tietokoneen suorituskyvyn tai kapasiteetin vaatimuksista ei löytynyt tarkkoja vaatimuksia tai suosituksia. Kehittäjä kuvaa järjestelmää webERP:n ohjekirjassa ”poikkeuksellisen kevyeksi” ja sen testattiin pyörivän pätevästi Educloudin kevyimmällä palvelimellakin yhdellä käyttäjällä. Ohjekirjassa mainittiin myös, että kehitystyössä tultaisiin jatkossakin ottamaan huomioon vähäinen resurssien tarve. Tosin dokumentaatio on vanha eikä sen ylläpidosta ole tietoa.

Oleelliseksi havainnoksi muodostui myös kehittäjän huomautus, että mitä useampia yhteyksiä (käyttäjiä) palvelimelle muodostetaan, sitä enemmän RAM-muistia olisi syytä löytyä järjestelmästä.

Koska palvelin tilattaisiin Educloud-palvelusta, tutkittiin palvelua tarkemmin Metropolian tietohallinnon sivuilta, jonka kautta opittiin, että valittavista kokoluokista small, medium ja large, tarjoavat järjestäen 1, 2 ja 4 gigatavua RAM-muistia. Tietohallinto suositti small-kokoa yleisimpiin tapauksiin, mediumia vaativimpiin kursseihin ja largea vain erikoistapauksiin. (Keski-Ruismäki 2018.)

Tarkka numeroihin perustuva analyysi tai tarvekartoitus resursseista olisi oletettavasti ollut turhan vaativaa laatia, etenkin suhteessa hyötyyn, joten tätä ei kartoitettu. Eräänä mahdollisuutena olisi voinut olla palvelimen tilasta mittausdatan tallentaminen käyttöjärjestelmän omien mittausohjelmien kautta, esimerkiksi sekunnin välein, ja sitten tutkia, miten eri käyttötilanteissa palvelin selviytyy kuormituksesta.

Jos halutaan tingata resursseissa, voitaisiin tilata kevyin palvelin ja havainnoida, selviytyykö palvelin aiotusta käyttötarkoituksestaan. Jos palvelin toimii hitaasti, voisi tilata isomman.

Teknistä lisätietoa

WebERP:n lähdekoodi sisältää PHP-kieltä. PHP on ohjelmointikieli, ja sen eräs mielenkiintoinen ominaisuus on, että PHP:llä tehty web-sovellus suorittaa palve-

lupyyntöjä palvelimen päässä, siinä missä esimerkiksi JavaScript-pohjainen sovellus tavallaan lataa suoritusohjeet palvelimelta ja sitten siirtää suoritusvastuuta loppukäyttäjän laitteelle.

Datan tallennukseen käytetään MySQL-tietokantaohjelmaa, joka tallentaa datan järjestelmässä. Käyttäjä ei tätä taka-alalla tapahtuvaa toiminnallisuutta havaitse käyttäessään selaimen kautta järjestelmää.

WebERP:hen liittyvä kirjoitusmuoto on alkuperäisen kehittäjänsä osalta kirjoitettu pienellä alkukirjaimella ja isolla loppuosalla, ja insinööriyön viime metreillä termin oikeinkirjoituksen osalta päätettiin konsultoida Kotimaisten kielten keskuksen asiakaspalvelua, joka antoi puhelimen kautta neuvontaa ja osoitti vielä lähdeviitteen keskuksen verkkoaineistoon kysymyksen osalta.

Asiakaspalvelu hyväksyi asiaa puitaessa kirjoituslogiikan, jossa poikkeuksellinen lainasana kirjoitettaisiin lauseen alussakin pienellä alkukirjaimella. Insinööriyössä valittiin mahdollisesti vaihtelevuudellaan lukijaa hämmentävä kompromissi, jossa lauseen alussa esiintyvä webERP päätettiin kirjoittaa isolla alkukirjaimella ja muutoin seurattiin kehittäjän lanseeraamaa nimeä. Loogisesti yhtenäisemmin voisi kirjoittaa webERP:n myös aina pienellä (tai aina isolla) alkukirjaimella, mutta kompromissilla pyrittiin kunnioittamaan suomen kieltä ja järjestelmän omaa kirjoitusmuotoa samanaikaisesti. (Kotimaisten kielten keskus.)

Käyttöönotto

Liike-elämän tilanteissa Kettunen ja Simons näkevät, että toiminnanohjausjärjestelmän käyttöönoton riskit voivat johtaa ”suuriin taloudellisiin menetyksiin ja estää toiminnanohjausjärjestelmään sisältyvien potentiaalien hyödyntämisen” (Kettunen & Simons 2001: 7).

Kettunen ja Simons mainitsivat tilanteesta, jossa pk-yritykset joutuvat tekemään kompromisseja ohjelmiston räätälöinnin (soveltavuuden) ja sen hinnan kanssa, jolloin usein tyydytään standardoituihin valmisratkaisuihin, jotka puolestaan eivät aina palvele liiketoiminnan ohjausta. (Kettunen & Simons 2001: 69.)

WebERP on ilmaista ohjelmistoa, joten käyttöönottoon ei kuulu siltä osin investointia. Sen lähdekoodi on maltillinen kooltaan ja sen omakohtainen jatkokehitys on mahdollista. Selainpohjaisten sovellusten (web-sovellusten) ohjelmoinnista käytetään termiä web-kehitys. Etu web-sovelluksissa on, etteivät ne ole laiteriippuvaisia. Järjestelmää voi käyttää eri laitteilla ja eri paikoista.

Haasteita

Arvioiden, haasteeksi webERP:n kohdalla muodostuu kommunikointi muiden yritysten järjestelmien kanssa, uusien ominaisuuksien kehitys ilman virallista tukea ja automaation kehittäminen.

”Ymmärtämättömyys toiminnanohjausjärjestelmän moninaisista vaikutuksista organisaation toimintaan liittyy pk-yrityksille tyypilliseen it-alan osaamisen puutteeseen” (Kettunen & Simons 2001: 69). Edellä mainitussa tilanteessa todennäköisesti tarvittaisiin ulkopuolista tukea järjestelmän käyttöönoton kanssa, ja avun löytäminen voi olla epävarmaa. WebERP:llä on omat keskustelupalstat, mutta aktiivisuus on ollut jossain määrin vähäistä. Myös webERP:n kotisivuilla (www.weberp.org) löytyy lista konsulttipalveluista, jotka tarjoavat ammattiapua webERP:n kanssa, mutta joiden työn laadusta ei ole takeita. Toisaalta vaikuttaa siltä, että itse järjestelmän kehittäjä tarjoaa myös konsultaatioapua.

WebERP:n kehittämistä ei ole hylätty, vaan päivityksiä on tullut koko ajan vuosien saatossa. Samalla ohjelmiston ylläpito on siirtynyt alkuperäisen julkaisijan kotisivuilta ja postituslistoilta GitHubiin.

WebERP:n käyttäminen yrityskontekstissa ilman ulkopuolista apua vaatii ymmärrystä omasta liiketoiminnasta ja osaamista toiminnanohjausjärjestelmistä sekä mahdollisesti ohjelmoinnista. Vaikka taidot olisivat kohdillaan, on toki oman ajan riittäminen käyttöönotto- ja kehitystyöhön pelkästään omin voimin arviointia vaativa seikka etenkin pk-yrityksessä, jossa henkilöresurssit on sidottu operatiiviseen toimintaan (Kettunen & Simons 2001: 116–117; 222).

3.3 Tietojärjestelmäkäsitteitä

Tässä alaluvussa käydään läpi insinööriyöhön ja tietojärjestelmiin liittyviä käsitteitä aakkosjärjestyksessä. Tarkoitus on lyhyesti esitellä joitain oleellisimpia teknologioita, joita liittyy webERP-palvelimen asentamiseen ja käyttöön.

Hallinnointi

Hallinnoijan rooliksi voisi nähdä palvelimen ylläpitoon liittyvät asiat, kuten käyttäjien hallinnoinnin, käyttöoikeuksien määrittämisen ja ohjelmistojen asentamisen.

On yleinen hyvä käytäntö, että palvelimen hallintaa varten tehdään erillinen tili, eikä käytetä juurikäyttäjän tiliä (root), jolla on täydet käyttöoikeudet. Työssä tehdään, hallinnointitilit käyttöjärjestelmälle (CentOS) sekä tietokantajärjestelmälle (MariaDB). Yhdistämme webERP:n tietokannan hallinnoijan tilin kautta, verraten että käyttäisimme juurikäyttäjää. Itse webERP:n sisälläkin on toiminnallisuus käyttäjien hallinnointiin, jossa voi myös luoda eri (yritys)roolin käyttäjiä, mukaan lukien järjestelmän hallinnoija.

Käytänteiden mukaisesti eri tileille olisi hyvä pystyä määrittämään niiden käyttötarkoitus ja antaa käyttöoikeuksia vain tarpeenmukaisesti. Hallinnoijalle annettaisiin suuremmat oikeudet kuin peruskäyttäjälle.

Palvelimelle voidaan yhteyden muodostamisen nopeuttamiseksi tallentaa kirjautumiseen käytetyn koneen tiedot palvelimelle, jolloin salasanaa ei tarvitse syöttää. Tällöin hallinnoijan tulee olla varma, ettei mikään luvaton taho käytä samaa laitetta yhteyden muodostamiseksi palvelimeen.

Komentotiedosto

Komentotiedostoa (skripti) kannattaa käyttää, kun järjestelmään tai ohjelmaan ajettavien komentojen määrä tai monimutkaisuuden taso kasvaa sekä kun halutaan automatisoida koneen toimintoja toistettavien käskyjen kautta. Skripti on

tiedosto, jossa ohjelmakoodi on valmiina suoritusta varten ja sitä voi lukea ja muokata tekstieditorilla. Skripti edustaa prosessia, joka on ennaltamääritetty ja tallessa myöhempiä, uudelleentoistettavia tilanteita varten, jolloin käyttäjän ei esimerkiksi tarvitse työläästi kirjoitella samoja komentoja joka kerta.

Insinööriyössä lukuisia, eri osa-alueista vastuussa olevia, skriptejä rakennettiin asennustyökalua kehitettäessä. Nämä eivät päätyneet insinööriyön ratkaisuun asti.

Hyvin skriptin tapainen ratkaisu laitettiin insinööriyön liitteisiin. Se onkin käytöltään hyvin samankaltainen, vaatien pienimuotoisen manuaalisen työn vaivan, jossa koko koodin maalaa ja kopioi sekä liittää komentotulkkiin palvelinta varten.

Komentotulkki

Komentotulkillä (toiselta nimeltä komentokehoite) viitataan tietokoneella olevaan ohjelmaan, joka suorittaa käyttäjän ohjelmaan antamia tekstisyötteitä käyttöjärjestelmän komentoina. Insinööriyössä avattiin komentotulkki omalta laitteelta, johon annettiin käsky ottaa yhteys palvelimeen. Tämän jälkeen komentotulkin kautta voidaan antaa komentoja palvelimelle.

Git Bash päätettiin ladata omalle Windows-laitteelle, sillä sen komentotulkki käyttää Linuxin syntaksia ja tunnistaa monia komentoja, pystyen ajamaan osaa niistä jopa Windows-koneella (verraten vaikka Windowsin komentotulkkieihin).

Tietohallinto on dokumentoinut toisen komentotulkin, PuTTY:n käytön (Keski-Ruismäki 2016). Bashin käyttöönottamiseen oltiin tyytyväisiä ja se tarjosi myös SSH-yhteyden palvelimelle, joten PuTTY:a ei päätetty käyttää, mutta se on vaihtoehto yhteyden muodostamiseksi.

Käyttäjätarina

Käyttäjätarinan (englanniksi user story) kautta hahmotellaan loppukäyttäjän näkökulmaa tietokoneohjelmaa suunnitellessa. Pohdinnassa on esimerkiksi ohjelman käyttäjän valmiudet ja motivaatio ohjelman käyttöä kohtaan. Tämä näkökulman hallinta pyrkii ohjaamaan ohjelman kehitystä loppukäyttäjä-, eikä kehittäjälähtöiseksi.

Käyttäjätarinassa pohditaan miten henkilö käyttäisi ohjelmaa, minkä pohjalta tunnistetaan kehitystarpeita, joita sitten nostetaan työn alle (esimerkiksi työliselle). Aitoja henkilöitä voi myös järjestää haastattelutilanteisiin ja livedemoihin.

LAMP

LAMP (LAMP stack, LAMP-pino) on valmiiksi koottu paketti ohjelmistoja, koostuen Linuxista, Apachesta, MySQL:stä ja PHP:stä. Insinööriyössä käytetään Educloud-palvelussa valmiiksi tarjottua LAMP-pinoa, jolloin tarvittavan ohjelmiston asentamista ei tarvitse itse aloittaa nollasta.

Linux

Linux on ilmainen, avoimen lähdekoodin käyttöjärjestelmä. Tarkemmin sanottuna Linux edustaa käyttöjärjestelmän ydintä, josta puolestaan on johdettu erilaisia jakeluversioita käyttöjärjestelmiksi. Insinööriyössä tilattavassa LAMP-pinossa Linuxia edustaa CentOS-jakeluversio.

Pienenä lisähyötynä opittiin, että Linux-järjestelmissä on itse käyttöjärjestelmän osalta pienempi RAM-muistin tarve kuin Windows-järjestelmissä.

MySQL (MariaDB)

MySQL on SQL:ää syntaksissaan käyttävä avoimen lähdekoodin tietokantajärjestelmä, josta palvelimella käytetty MariaDB periytyy. Nämä järjestelmät ovat

näkökulmastamme samoja, sillä ne tunnistavat saman syntaksin ja samat käskyt. Tietokantajärjestelmää käytetään nettipalvelimella datan hallintaan.

Kätevää (tietohallinnon tarjoamassa) valmiissa LAMP-pinossa on, että siinä on myös tietokannan juurikäyttäjä (root) alustettu. Tämä tarkoittaa juurikäyttäjälle on asetettu salasana, joka muuten puuttuisi tuoreesta asennuksesta ja salasana pitäisi määrittää itse. Lisäksi tietokantaa (kuten itse käyttöjärjestelmää) varten on hyvä käytäntö tehdä hallinnollinen tili.

Nettipalvelin

Nettipalvelin on ohjelmisto, jonka vastuulla on suorittaa selaimen kautta annettuja URL-pyyntöjä. Insinööriyössä käytettiin Apachen nettipalvelinohjelmistoa.

Docroot (document root) on nettipalvelimen kotikansio nettiselaimen kautta koettavien palveluiden tiedostoille. Docrootissa sijaitsee siis nettisivut tallennettuina.

Insinööriyössä valittiin palvelin LAMP-pinolla, jossa on Apachen tarjoama nettipalvelin valmiiksi asennettuna. Näin ollen kotikansio (docroot) on valmiiksi määritetty palvelinta käyttöönotettaessa.

WebERP päätettiin asentaa nettipalvelimen kotikansioon. Vaihtoehtoisesti asennuksen voisi määrittää omavalintaiseen sijaintiin ja konfiguroida nettipalvelin viittaamaan oikeaan polkuun. Sitä, parantaisiko epätavallinen asennuspolku palvelimen tietoturva ja kuinka marginaalisesti, ei alettu selvittämään. Hallinnoijan kannalta tilanne olisi epäselvempi, kun tiedostorakenne olisi epälooginen.

PHP

PHP on ohjelmointikieli, jonka avulla ohjelmoijaa rakentamaan selainten tulkitsemää HTML-koodia tehokkaasti. Sen avulla luodaan esimerkiksi yhteyksiä web-sovellusten ja tietokantajärjestelmien, kuten insinööriyössä käytettyjen webERP:n ja MariaDB:n, välille.

SSH

SSH on työkalu, jonka avulla muodostetaan kryptattu kommunikointiväylä toiseen laitteeseen. Väylää käytetään etänä työskennellessä. Esimerkiksi insinööriyössä muodostettiin yhteys kotikoneelta Metropolian palvelimelle.

Kryptaaminen tarkoittaa tässä sitä, että yhteyden yli välitettävä tieto, joka saattaa sisältää salassa pidettävää informaatiota, muutetaan muotoon, josta sitä ei voi luontaisesti lukea, eikä sitä ole helppo arvaamalla muuttaakaan takaisin luettavaan muotoon. Kryptausta pidetään tietoturvan kannalta oleellisena asiana, sillä vaikka jokin taho kykenisi seuraamaan laitteiden välistä yhteyttä luvatta, pysyisi tieto silti äärimmäisen todennäköisesti salassa.

Ladatulle Git Bashin ja sen komentotulkin mukana tulee myös SSH, jota on yksinkertainen käyttää. Tietohallinto on luonut puolestaan ohjeet SSH-tunneloinnista PuTTY-ohjelmaa käyttäen, mistä mainittiin aiemmin otsikon ”komentotulkki” kohdalla.

Työlista

Työlistalla (englanniksi backlog) tarkoitetaan joukkoa ominaisuuksia, jotka olisi tarkoitus saada tuotua eloon kehitettävään ohjelmaan koodinsa kautta. Sitä voi ajatella (projektin) työvaiheiden koontina, josta sitten valitaan osasia työstettäväksi. Ominaisuuksia voi listata tärkeysjärjestykseen, painottaen niiden valmistumista ennen vähemmän kiireellisiä osasia.

3.4 Palvelimen resurssitarpeet

Insinööriyön ensimmäisenä suoritteena arvioitiin sopivan ja oikeinmitoitettun palvelimen hankintaa. Hankinta puolestaan tehtäisiin Metropolian palvelukatalogista, joka löytyy Educloud-nimisestä palvelusta.

Kapasiteetin arvioimista varten tutkittiin palvelimien keskimääräisiä tarpeita eri lähteistä, minkä ajateltiin antavan vertailupohjaa arvioidessa webERP-palvelimen oikeaa mitoitusta. Toimeksiantajan kanssa käydyssä sähköpostiviestityössä arvioitiin käyttäjien määräksi noin kolmisenkymmentä samanaikaista käyttäjää.

Yleisesti ottaen sopiva palvelimen kapasiteetin mitoitus välttää tarpeetonta ylimitointia, jolloin liiallisista resursseista maksettaisiin tyhjää, mutta huomioi, että kapasiteetti riittäisi palvelimen pyörittämiseen riittävällä käyttökokemuksella.

Palvelimen skaalaus on eräs asia, jonka voisi ottaa huomioon elinkaarta suunniteltaessa, jos yrityksen toiminnan ja datan määrän odotettaisiin kasvavan ajan saatossa. Opetuskäyttöä ajatellen tätä ei tarvitse huomioida lainkaan, sillä palvelimilla on muutenkin ajallisesti rajattu ja lyhyehkö elinkaari (pois lukien erikoistapaukset).

Kaupallisia toiminnanohjausjärjestelmiä

Internethauilla löydettiin erilaisiin käyttötarkoituksiin valjastettujen palvelimien suorituskyvyn ja kapasiteetin laitteistovaatimuksia. Hakuja kohdennettiin myös, asiaan liittyen, kaupallisten toiminnanohjausjärjestelmien laitteistovaatimuksiin. Kokonaisuudesta ei kuitenkaan saanut avartavaa ja selkeää kuvaa, vaan jäi käsitys, että eri ohjelmistot (asiayhteydestä puhumattakaan) omaavat eri kokoluokan vaatimuksia: osan toiminnanohjausjärjestelmistä havaittiin olevan niin raskaita, ettei Educloudin tehokkainkaan palvelinpaketti riitä täyttämään niiden minimisuositusta.

Esimerkkisuositus 100 gigatavun tallennustilan palvelimelle – joka olisi tilantarpeeltaan reilusti yli kaksinkertainen Educloud-katalogista kapasiteetiltaan löytyville vaihtoehdoille – oli suorituskykyinen prosessori 8 ytimellä ja 64 gigatavulla RAM-muistia. (insightsoftware.)

Puhtaasti spekuloiden voisi arvella, että bisnesohjelmistojen raskauteen vaikuttaa ison datan tallentamisen lisäksi juuri saman datan analysointiin vaadittava teho ja kenties graafiset käyttöliittymät. Näiden käyttötarkoitus lienee suunnattu-kin pientä yritystä suurempiin organisaatioihin.

Arviointityökalu

Kartoituksen lomassa löydettiin työkalu, joka arvioi MySQL-palvelimien tarpeita (MySQL Calculator). Kyseisen työkalun käyttäminen vaatisi toisaalta tarkempaa tietämystä siitä, mitä palvelimella aiotaan tehdä. Oletettavasti opetustarpeeseen resurssien tarve jäisi maltilliseksi. MySQL:n omassa dokumentaatiossa mainitaan, että yleensä pärjätään hyvin pienelläkin määrällä RAM-muistia. Dokumentaatiossa annettiin muistin allokointivinkkejä 256 MB:n sekä 1–2 GB:n kokoonpanoille. (Oracle: Configuring the Server).

Skaalaamisesta

Yritysten elinkaarissa esimerkiksi toiminnan kasvaminen saattaa heijastua myös tietojärjestelmän puolelle kapasiteetin riittämättömyytenä. Insinööriyön konteksti on luonteeltaan rajatumpi, kun elinkaari on jo palvelukatalogissa rajattu muutama kuukauteen. Jatkuvassa yritystoiminnassa dataa kumuloituu talteen vuosien saatossa.

Kapasiteettia vaativia tekijöitä

Jos webERP:tä haluttaisiin muokata kehittämällä sen lähdekoodia esimerkiksi tuomalla uutta toiminnallisuutta järjestelmään, voisi tällä olla vaikutusta kapasiteetin tarpeen kannalta. Laajennettu järjestelmä tarvitsee myös enemmän resursseja.

Käyttäjämäärät voivat yleensä vaikuttaa datan paljouteen ja palvelimen kuormitukseen, mutta esimerkiksi opetustilanteessa voitaneen aavistaa etukäteen, ettei palvelimen elinkaaren aikana oletettavasti palveltaisi enempää kuin korkeintaan kymmeniä samanaikaisia käyttäjiä. Suurempi määrä samanaikaisia

käyttäjää vaatii enemmän resursseja, mutta kymmenet käyttäjät eivät liene ongelma. Liiallisen kapasiteetin varaaminen maksaa ja tässä kontekstissa ollaan paremmin mielin liikkeellä skaalaamisen suhteen kuin jos pitäisi tehdä päätöksiä voimakkaasti vaihtelevan tarpeen äärellä.

Suorituskyvyn vaatimuksia periaatteessa nostaa webERP:n käyttämä PHP, jonka osalta palvelin suorittaa itse selaimen kautta tulleita pyyntöjä. Kevyemmän ratkaisun saattaisi muodostaa esimerkiksi JavaScript, jossa (joitain) palvelupyntöjä suoritetaankin loppukäyttäjän laitteella palvelimen sijaan. Mutta webERP itsessään on todella kevyt, joten pienimuotoisessa käytössä eroavaisuuksia ei liene käytön sujuvuuden suhteen.

Windows-pohjaiset palvelimet vaativat enemmän RAM-muistia kuin Linux-pohjaiset, joten tämä näkökulma tuki Linux-palvelimen valintaa.

3.5 Näkökulmia tietoturvaan

Tietoturva-asioihin kantaa ottaminen oli eräs insinööriyön tavoite. Tilanne insinööriyössä tuntuisi tietoturvan kannalta olevan mukavasti hoidossa jo lähtökohteisesti.

Koski toteaa aiheesta, että täysin turvallinen järjestelmä on mahdottomuus. Hän myös jakaa Linuxin tietoturvan sisäiseen ja ulkoiseen osaan, missä viitataan järjestelmän sisäisiin käyttäjiin ja sen ulkopuolisiin käyttäjiin sekä näiden muodostamiin tahallisiin tai tahattomiin tietoturvauhkiin. (Koski 2010: 483.)

Sisäisen turvallisuuden pohdintaa

Sisäisen turvallisuuden kohdalla huomio on järjestelmän käyttäjissä. Inhimillisiä virheitä voi sattua eri tavoin. Sähköpostilla saapuneiden väärin linkkien ja tiedostojen ajaminen on antanut hakkereille jopa voimallaitoksia käsiinsä maailman historiassa.

Insinööriyössä ei pohdita henkilökohtaista tai organisaation tietoturvakulttuuria sen enempää. Tälle ei välttämättä olisi juuri tarvettakaan, sillä – käyttötarpeesta riippuen – palvelimen käyttöjärjestelmään ei tarvinne tehdä mitään ylimääräisiä tilejä muille käyttäjille, sillä webERP-palvelimen datasta huolehtii LAMP-pinon mukana tullut ohjelmisto ja käyttäjille tilanne realisoituu webERP:n sisäänkirjautumissivulla; sen käyttäminen vaatii tunnuksen tiedot tietokantaan (ei niinkään palvelimen käyttöjärjestelmään).

WebERP:n käytön kannalta hallinnoijalle eräs inhimillisen virheen paikka voisi olla antaa järjestelmää käytettäessä ja käyttäjiä siihen luotaessa järjestelmän sisäiset hallinnointioikeudet jollekin epäluotettavalla käyttäjälle, joka sitten kykenisi esimerkiksi häiriön aiheuttamiseksi poistamaan dataa tietokannasta.

Ulkoisen turvallisuuden pohdintaa

Ulkoinen turvallisuus viittaa järjestelmän ulkopuolelta tulleisiin uhkiin. Käytännössä halutaan siis rajata ulkopuolelta tulevia yhteyksiä järjestelmään.

Tässäkin mielessä voidaan merkittävästi huokaista helpotuksesta. Itse Educloudin kautta tilattavat palvelimet ovat rajattu toimimaan vain Metropolian verkon kautta, mikä lisää tietoturvan tasoa paljolti. Vaikka ulkopuolinen tietäisi webERP-palvelimen osoitteen, hänen pitäisi ensin pystyä pääsemään Metropolian verkkoon sisään. Lisäksi vaikka ulkopuolinen tietäisi webERP-palvelimen käyttöjärjestelmän juurikäyttäjän tunnuksen ja salasanan, ei hän siltikään pääsisi palvelimelle, sillä hänen IP-osoitteensa olisi estetty.

Tietohallinnon ohjeissa on opastettu VPN:n käytöstä, jota käydään asennusohjeissa läpi lisää alaluvussa ”palvelimen tilaaminen”. Tämän kautta pääsee webERP-palvelimeen muodostamaan yhteyden muualtakin kuin Metropolian toimiloista; insinööriyökin tehtiin etänä tätä väylää käyttäen.

Yleisiä käytänteitä

Tietojärjestelmissä eräs peruskäytäntö on antaa käyttöoikeuksia vain tarpeen mukaan. Insinööriyössä kohdattiin esimerkiksi tilanne, jossa webERP ei saanut tietokantajärjestelmään yhteyttä. Tämä johti prosessiin, jossa nettipalvelimelle annettiin täydet oikeudet ja sen jälkeen iteroitiin asiaan kuuluvia käyttöoikeuksia aina pykälä kerrallaan pienemmiksi. Lopulta todettiin, että ajamisen oikeuksia ei tarvittu, luku- ja muokkaus-oikeudet riittivät.

Myös järjestelmän hallinnointiakin rajoitetaan; käyttöjärjestelmän juurikäyttäjän tiliä ei suositella käyttämään, ellei sitä nähdä tarpeelliseksi. Tästä syystä järjestelmään on tapana luoda hallinnointia varten oma tili, ellei useampaa. Asennusohjeissa asennus tehdään kuitenkin käyttöjärjestelmässä juurikäyttäjällä, kun taas myöhemmin webERP ohjataan (tietokannan) hallinnoijan tiliin. Juuren käytön perusteluna on lähinnä oikoteiden hakeminen ja se, että ohjeet ovat vakiot, mikä karsii yllättävät asiat pois, jos ohjeissa pysytään.

Kun kommunikointiväylä omalta laitteelta palvelimelle on muodostettava etäyhteydellä, käytetään salattua väylää. SSH:sta mainittiin alaluvussa 3.3. Potentiaalinen riski on muutoin, että lähetetty tieto voidaan kaapata ja mikäli se on luontaisessa ja luettavassa muodossa, voi tietoliikennettä salakuunteleva taho saada käsiinsä salassa pidettävää tietoa.

Tietokantaan liittyviä asioita

Puhdas asennus jättäisi tietokantajärjestelmän juurikäyttäjän tilin alustamatta, jolloin juurikäyttäjän tilillä ei ole asetettu salasanaa. Tässä mielessä Educloudin valmis LAMP-pino on myös kätevä ratkaisu, sillä tietokannan juurikäyttäjälle on asetettu salasana valmiiksi.

Tietokantajärjestelmän hallinnoijan tili määritetään toimimaan localhost-arvolla. Vaihtoehtoisesti "%" -arvo hyväksyy yhteyden muodostamisen mistä vain. Tätä ominaisuutta emme tarvitse ja siihen voi liittyä tietoturvariski, tosin yhteyksiä suodatetaan jo palvelimen toimittajan puolesta. Localhost on tietoturvallisempi,

koska silloin tietokannan hallinnoijaa voi käyttää vain suoraan palvelimelta, joten tietomurtajan tulisi ensin päästä palvelimen käyttöjärjestelmään.

Näkemyksiä yritysten toiminnanohjausjärjestelmien tietoturvasta

Tivin artikkelissa (Savonen 2020) KPMG:n tietoturvajohtaja Mika Laaksosen huomioista saa käsityksen, että yrityksillä on pääosin toiminnanohjausjärjestelmien tietoturva tunnollisesti huomioituna. Laaksosen kuitenkin muistuttaa, ettei laadukkaallakaan suojalla ei ole väliä, mikäli tietomurtaja pääsee iskemään inhimilliseen erheeseen.

Tietojen kalastelussa järjestelmän käyttäjä kohtaa esimerkiksi väärennetyn pyynnin kirjautua johonkin järjestelmään tai palveluun. Kirjautuminen tapahtuu-kin sitten tietomurtajan hallitseman prosessin kautta.

Samassa artikkelissa SAP:n yritysarkkitehti Eero Koskinen mainitsee, että toiminnanohjausjärjestelmät ovat houkuttelevia niiden suuren datamäärän vuoksi. Hänkin tukee Laaksosen ajatusta, että tietoturvan tilanne [Suomessa] on hyvä ja kertoo syyksi sen, että tietoturva usein ulkoistetaan [ammattimaisille tahoille].

3.6 Projektinhallinnallisia seikkoja

3.6.1 Projekti

Projektin määrittelemisen terminä tuntuu olevan jossain määrin taiteenlaji, eri määritelmien paljouden takia. Artto ym. määrittelivät projektin lainauksen arvoisesti:

Projekti on ennalta määritettyyn päämäärään tähtäävä, monimutkaisten ja toisiinsa liittyvien tehtävien muodostama ajallisesti, kustannuksiltaan ja laajuudeltaan rajattu ainutkertainen kokonaisuus. (Artto ym. 2008: 26.)

Projektin ainutlaatuisuus nähdään tässä insinööriyössä siinä, että työn tavoitetta ei ole mielekäästä havitella useaa kertaa. Toisin sanoen pyörää ei ole järkevää keksiä uudelleen. Parantaminen onkin sitten eri asia.

Ennen kaikkea pyörä on tehty ajamista, eli prosessia, varten. Prosessi on luonteeltaan (tarkoituksenmukaisesti useasti) toistettava (ellei jopa jatkuva) työrypeama, kuten ajomatka autolla kodista töihin.

Tämä insinööriyö voidaan nähdä projektina, jossa määritettiin prosessi webERP-palvelimen pystyttämiseen (tilaaminen ja asennus), Metropolian palveluita käyttäen.

Projektikolmio kuvassa 3 on yleinen projektinhallinnallinen käsite, jolla havainnollistetaan yllä annetun projektin määritelmän rajoitteita. Projektiin ja sen läpimenoon liittyvät rajoitteet ovat tavoitteet, aika ja kustannukset. Nämä elävät jatkuvasti suhteessa toisiinsa nähden.



Kuva 3. Projektikolmio muistuttaa projektin rajoitteiden välisistä suhteista.

Esimerkki valaisee asiaa helpoiten: jos projektista halutaan enemmän irti ja sille asetetaan suurempia tavoitteita, sen läpimenoon menee myös suurempi aika – ellei projektiin allokoita lisää resursseja, jolloin puolestaan kustannukset kasvavat. Tai voidaan ajatella, että tiiviimpi ja vähemmän mahtipontinen projekti on

vastaavasti nopeampi toimittaa, tai sitten se valmistuu samassa ajassa, mutta vähemmällä resursseilla.

Insinööriyön aikana haluttiin etsiä väylää työmenetelmien tehostamiseen. Päätettiin tutustua lisää jo opintojen aikana alustavasti tutuksi tulleisiin ketteriin menetelmiin. Muitakin aihepiirejä tutkittiin, mutta työhön päätyen kaiken dokumentoiminen olisi kasvattanut projektia jo turhastikin.

Kirjallisuuden tarkastelun näkökulmana pyrittiin luontaisesti pitämään insinööriyön konteksti mielessä, jossa erityispiirteenä oli se, että työ tehtiin itsenäisesti, verraten ryhmätyöhön, jossa useamman henkilön työresurssien hallinta on oleellisempaa.

Niin syvälliseen analyysiin ei ryhdytty, että työn tekemisen tehokkuuden tarkastelua varten olisi laadittu mittaristoa. Project Management Institute (jatkossa PMI) esitteli yksinkertaisena mittarina, että ketterien menetelmien projekteissa voidaan mitata (tuotantoon) valmistuneita työvaiheita kumulatiivisesti aikaa vastaan. Ennakoitavissa projekteissa voidaan mitata, esimerkiksi kuinka monta metriä jotain elementtiä on asennettu tähän mennessä ja kuinka paljon on asennattamatta, mutta muuttuvissa projekteissa tarve elää projektin mukana, mikä tekee ennakoimisen vaikeaksi.

Edellä mainittua mittaria voidaan käyttää ketterästi työskentelevän tiimin työ- tai toimitustahdin arvioimiseen, jonka kautta saataisiin käsitystä työtehosta ja täten koko projektin kestoa voitaisiin arvioida valistuneemmin.

PMI näkee (2017: alaluku 5.4), että työn valmistumisen mittaaminen olisi tärkeämpää kuin työvaiheen tilanteen mittaaminen, sillä ketterissä menetelmissä on tarkoitus saada toimitettua arvoa tuotantoon. Lisäksi työvaiheen tilanne voi heitellä rajusti, mistä käytetään termiä vesimeloniprojekti (englanniksi watermelon project), joka kuvastaa sitä, että työvaihe on nähtävästi vihreä (kaikki kunnossa ja valmiina tuotantoon), mutta paljastuukin pintapuolisen tarkastuksen jälkeen yllättäen punaiseksi (riittämättömäksi tai väärin tehdyksi tarpeisiin nähden).

Insinööriyö sisälsi selvitystyötä ja ongelmanratkaisua. Näihin piirteisiin ketterät menetelmät sopivat olemukseltaan hyvin, kun kerran työssä edetään kohti ennalta tuntematonta ratkaisua.

3.6.2 Ketterät menetelmät

Ketterät menetelmät eivät luo kyvykkyyttä tuottaa lisää työtä, mutta niiden rooli on hallita arvon tuottamista, lyhyempien työrupeamien pilkkoessa kokonaisuutta osiin (Project Management Institute 2017: alaluku 5.4.1). Myös ketterien menetelmien manifestissa mainitaan pyrkimys tiheätahtiseen ja käyttövalmiiseen arvon toimittamiseen (Beck 2001: Principles behind the Agile Manifesto).

Ketteristä menetelmistä mainittiin mielenkiintoisena näkökulmana niissä olevan kyse tekemättömän työn maksimoinnista ja tämän painotettiin olevan oleellista (Beck 2001: Principles behind the Agile Manifesto). Tämän voisi johtaa tarkoittavan sitä, että jos jokin ei palvele kokonaisuutta ja pienintä riittävän hyvää ratkaisua, sitä ei ehkä tulisi toteuttaa.

PMI:n ilmaisun mukaan, projektityö voidaan jakaa määriteltävään sekä suuren epävarmuuden työhön (Project Management Institute 2017: alaluku 2.1). Määriteltävyys tarkoittaa sitä, että työn kokonaisuus voidaan nimenomaisesti määrittää – yleensä etukäteen kokemuksen kautta – tarkasti; esimerkiksi tietyn työvaiheen kestoaikea tai vaadittavat resurssit ovat ennustettavissa. Ennustettavuutta tukee se, että työvaiheessa on pieni riski yllättäviin muutoksiin, kuten konerikkoihin.

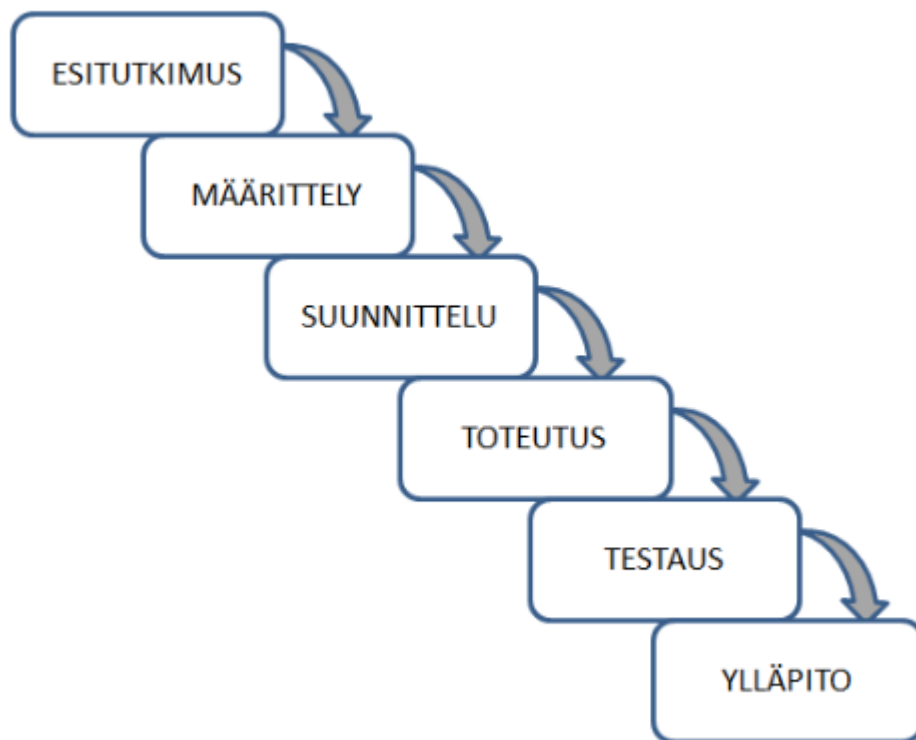
Vastaavasti epävarmuutta projektin läpimenoa tuovat, tavoitteiden muuttuminen projektin edetessä, kompleksisuus ja erilaiset riskitekijät (Project Management Institute 2017: alaluku 2.1).

Johdantona ketterille menetelmille käsitellään, vertailun vuoksi, lyhyenä alustuksena perinteisiä menetelmiä projektinhallintaa kohtaan.

Perinteiset menetelmät

Perinteisten menetelmien kautta projektin hallinnoiminen on otollista projekteissa, joissa työvaiheet ovat (etukäteen) helpommin määriteltävissä. Projektille saadaan lineaarinen ja keskeyttämätön jatkumo alusta loppuun, jos mitään yllättävää ei tapahdu.

Vesiputousmalli kuvassa 4 on esimerkki perinteisiin menetelmiin kuuluvasta projektinhallinnan mallista, jossa projekti "virtaa" työvaiheesta toiseen, kunnes saavuttaa päätöksensä. Projektin suunnittelussa pyritään olemaan huolellisia ja kattavia, sillä virtaavan jatkumon katkeaminen tarkoittanee epätoivottuja muutoksia projektin elinkaareen.



Kuva 4. Esimerkki vesiputousmallista (lähde: Tomi Männistö, opinnäytetyö, 2015: 8).

Opintojen aikana jossain määrin tutuksi tulleessa SAP:n kehittämässä, ja samalla nimellä tunnetussa, toiminnanohjausjärjestelmässä oli esimerkiksi mahdollisuus kuvata työvaiheiden kesto- tai läpimenoaikoja. Mikäli projektin sisältämät työvaiheet, sekä suoritteiden työjärjestys, tunnettaisiin läpikotaisin ennen projektin alkamista, olisi projektin suunnittelu työvaiheiden järjestyksen asettamista ja teoriassa vain jokseenkin triviaalia optimointia.

Projekteihin liittyy kuitenkin riskejä, jotka ovat enemmän tai vähemmän yllätyksellisiä luonteeltaan. Kaikkia työvaiheita ei välttämättä tunneta, tai osata riittävän tarkasti arvioida, projektia suunnitellessa. Lisäksi tavoitteet saattavat muuttua kesken projektin.

PMI kritisoi perinteisiä menetelmiä huomauttamalla, että vaikka projektin edistyminen olisi projektisuunnitelmaan nähden näennäisesti jopa lähes valmiskin, voi projektista yllättäen paljastua realisoituvia riskejä. (Project Management Institute 2017: alaluku 5.4.)

Ketterät menetelmät

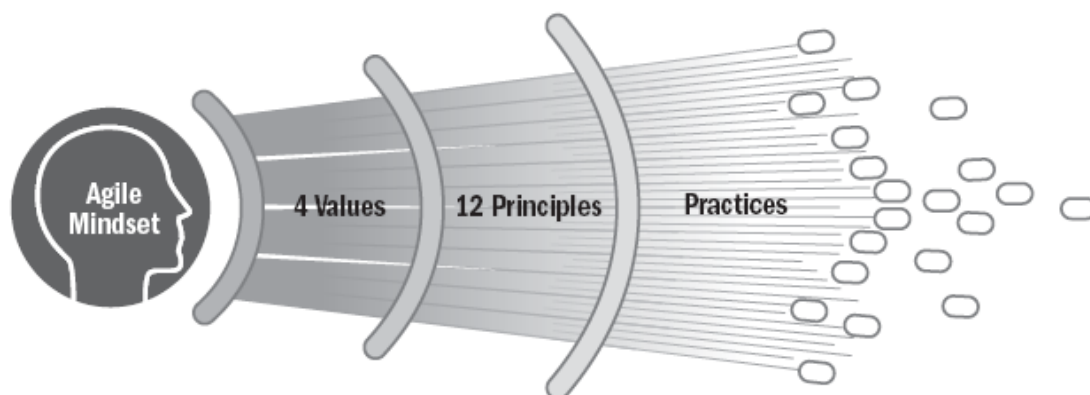
Tutkimustyö on hyvä esimerkki työstä, joka sisältää vaikeasti etukäteen määriteltäviä työvaiheita, sillä siinä määritelmän mukaan pyritään oppimaan, löytämään tai tekemään jotain uutta. On esimerkiksi vaikea ennustaa, kuinka kauan kestää löytää vastauksia tai ratkaisuja projektissa esitettyihin ongelmiin.

Ketterät menetelmät pyrkivät vastaamaan sekä ennalta tuntemattomien että muuttuvien työvaiheiden tarpeisiin paremmin kuin perinteiset menetelmät.

Ketteriä menetelmiä, tai niistä johdettuja käytänteitä, on monenlaisia. Insinöörityössä kattavaa yleiskuvaa ei pyritty muodostamaan. Sen sijaan pyrittiin tutustumaan joihinkin peruskäsitteisiin sekä malleihin.

PMI havainnollistaa ketterien menetelmien käytännön tason käytänteiden periytymisen ketterän manifestin mukaisista, 12 periaatteesta, jotka puolestaan periytyvät 4 perusarvosta (Project Management Institute 2017: alaluku 2.2; Beck

ym. 2001: Manifesto for Agile Software Development). Kuvassa 5 välittyy edellä mainitun periytyvyyden lisäksi menetelmien paljous.



Kuva 5. Erilaisten ketterien käytänteiden paljous ja periytyminen yhteisistä periaatteista sekä arvoista (muokattu lähteestä: Project Management Institute 2017: kuva 2-3).

Peruskysymys oli, auttaisiko ketterien menetelmien soveltaminen insinööriyötä etenemään tehokkaammin vai ei. Kontekstin puitteissa kysymys täsmentyi tarkemmin koskemaan sitä, voiko ketteristä menetelmistä olla hyötyä yhden hengen projekteissa. Ironiselta kuulostaen erilaisissa lähteissä puhuttiin yhden hengen tiimeistä.

PMI näkee, että ketterien menetelmien räätälöinti vaatii asiantuntemusta onnistuakseen (Project Management Institute 2017: liite X2.2). Työn aikana todettiin, että lienee syytä pysytellä valmiissa menetelmissä.

Keskustelufoorumeilta löytyi kattaus erilaisia mielipiteitä, joista (pienehköllä otannalla) muodostui käsitys, että työn visualisointi voisi olla jokseenkin yleispätevästi hyödyllistä. Tässä hyötynä olisi etenkin työn etenemisen hahmottaminen.

Yleisiä suosituksia keskustelupalstoilla olivat muun muassa: henkilökohtaiset muistiinpanot, koodin dokumentaatio, sekä ohjelmiston kehittämiseen liittyvän työlistan muodostaminen.

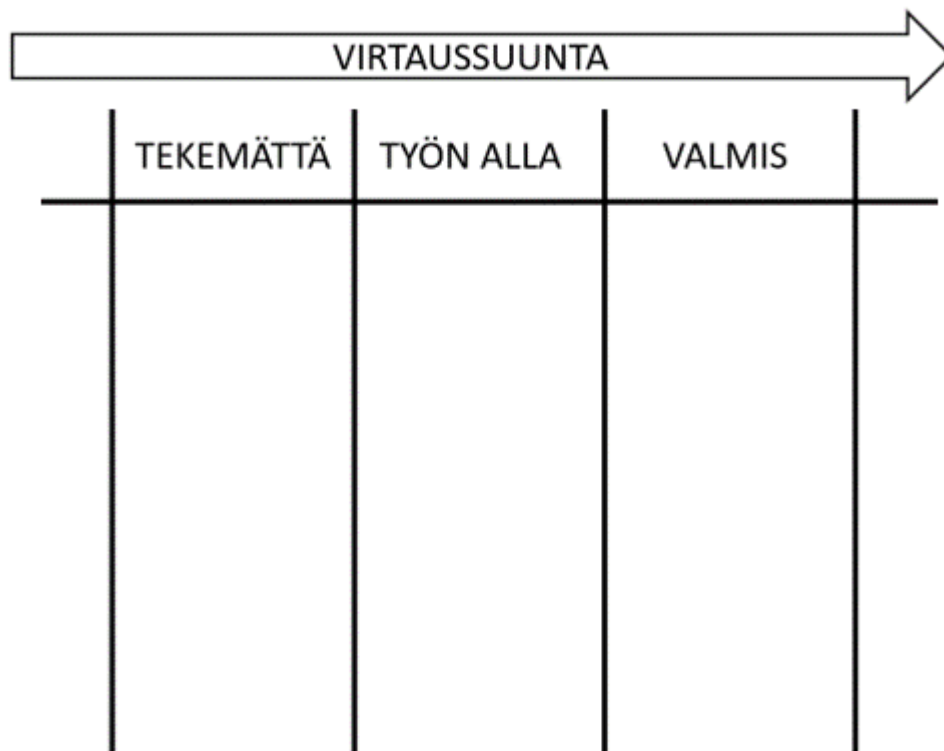
Työn visualisoinnin osalta ryhdyttiin tutkimaan Kanban-nimistä menetelmää. Kanbanista lisää seuraavaksi.

Kanban

Teollisesta Lean-tuotannosta ammentava Kanban ilmestyi 2000-luvun ensimmäisellä vuosikymmenellä vaihtoehdoksi aikakautensa ketterille menetelmille (Project Management Institute 2017: alaluku 2.3).

Hietaniemi näkee Kanbanin olevan Scrumin kanssa toinen tunnetuimmista ketterien menetelmien työkaluista sekä ilmaisee sen soveltamisalaksi minkä tahansa työn tehokkuuden parantamisen. Kanban toimii lähtökohtana prosessin kehittämiseksi, muttei anna valmiita vastauksia. Loppukaneetiksi Hietaniemi näkee jatkuvan kehittämisen olevan keskiössä ja kehoittaa kokeilemaan omanlaisen Kanban-työkalun rakentamista ilmenevään tarpeeseen sopivaksi. (Hietaniemi 2020.)

Kuvassa 6 on esimerkki yksinkertaisesta Kanban-työkalusta ja siitä käy perusidea selväksi: työtä kategorisoidaan sarakkeittain sen tilanteen mukaan. Työkalulle laitetaan (yleensä) lappuja, jotka kuvastavat yksittäistä työkohdetta.



Kuva 6. Yksinkertaisen Kanban-taulun runko. (muokattu lähteestä: Hietaniemi 2020)

Hämäläinen näkee Kanbanilla kolme pääkohtaa: työn visualisoinnin, samanaikaisen työn rajoittamisen ja uuden työn vetämisen (tarpeeseen perustuva imuohjaus) työntämisen sijaan. Korkeimpana näkökulmana Hämäläinen näkee, että arvoa muodostetaan vain valmistuneella työllä. (Hämäläinen 2016; Hämäläinen 2017.)

Myös Hammarberg ja Sunden painottavat visualisoinnin tärkeyttä. Tarkoituksena on luoda kuva Kanban-taulun muodossa projektista työvaiheineen silmällävään muotoon, jolloin sen nykytilan voisi hahmottaa kätevästi. Työn visualisointi kuvastaa työn virtausta aloituksesta valmistumiseen. Se auttaa hahmottamaan kuinka paljon työtä yhteensä on olemassa, kuinka paljon siitä on työn alla ja kuka tai mikä taho työstää mitäkin vaihetta (Hammarberg & Sunden 2014: alaluku 1.2).

Työn tehokkuuden mittaamisen kannalta esitettiin kaksi yksinkertaista mittaria: "lead time", jossa mitataan yksittäisen työkohteen läpimenoaikaa prosessissa, ja "throughput", jossa mitataan lukumäärällisesti, kuinka monta työkohdetta työntekijä saa suoritettua tietyn ajan kuluessa. (Hammarberg & Sunden 2014: alaluku 1.8.)

Mittarituloksia taas voisi mahdollisesti käyttää prosessinparantamiseen esimerkiksi niin, että työn rajaamiseen otetaan erilainen ratkaisu käyttöön ja sitten verrataan, muuttuuko esimerkiksi throughput muutoksen jälkeen; alkaako työntekijä saamaan enemmän työkohteita valmiiksi samassa ajassa.

Mittaamisen osalta insinööriyössä arvioitiin, ettei mittariston muodostamisesta saisi tarpeeksi hyötyä suhteessa vaivannäköön. Varsinkin kun mittaamista pitäisi soveltaa, sillä Hammerberg ja Sunden puhuivat kahden viikon mittausajanjaksoista. Insinööriyön aikana tuli taukoja muutenkin.

Personal Extreme Programming

Keskustelufoorumeista löydettiin myös vinkki menetelmästä nimeltä Personal Extreme Programming (kansainvälisesti tunnettu lyhennysmuoto on PXP, jota käytetään jatkossa). Se on ketteriin menetelmiin pohjautuva ohjelmistokehitysmenetelmä, joka on suunniteltu yhden hengen tiimeille (vaikkakin yhden henkilön ryhmä kuulostaa ironiselle).

Agarwal ja Umphress (2008) kertovat kehittämänsä PXP-menetelmän perustuvan Extreme Programming sekä Personal Software Process -menetelmiin. Menetelmä tarjosi konkreettisen ja askeleittain muodostetun kehitysprosessin, mutta se oli turhankin tekninen ja laaja opiskeltavaksi syvällisemmässä mielessä.

Prosessi jaetaan yleisesti suunnittelun ja toteutuksen vaiheisiin. Suunnittelu nojaa käyttäjätarinoiden luomiseen, joiden pohjalta sitten muodostetaan koodin menevää toiminnallisuutta. Toteutusvaihe on kirjattu hyvin koodimaiseen, ehtorakenteelliseen prosessimuotoon.

Minimum Viable Product

Minimum Viable Product (jatkossa MVP) on saanut suomeksi erilaisia nimiä, kuten ”pienin markkinoitava ominaisuusjoukko” tai ”pienin toimiva tuote”. MVP määritettäköön eräänlaisena laadun rajoittimena, jonka on tarkoitus rajoittaa suoritettavan asian tai projektin tavoitteita (laatua), jotta saataisiin parempi valmistumis- tai läpimenoaika. Tilannetta voi verrata aiemmassa aliluvussa mainittuun projektikolmioon (kuva 3).

Eräs MVP:n sovellus voisi olla keskittyminen karsimaan epäoleellisia asioita pois paisuneilta työlistoilta, jotta saataisiin nopeammin alkukantainen versio työstä toimitettua. Kyseessä on sen määrittelemisen, mikä olisi riittävä, ”pienin toimiva” tuote.

Asennustyökälua rakennettaessa oli työlistalla paljon työstettävää, eikä sitä olisi milloinkaan ollutkaan kokonaisuudessa järkevää toteuttaa ilman merkittävää asioihin paneutumista. MVP antoi hyvää suuntaa nopeaa kehittämistä varten. Aluksi haluttiinkin demonstroida ohjelman toimivuutta mock-up -henkeen, missä mock-up viittaa tässä toiminnallisuudeltaan rajattuun versioon, jossa painopiste oli enemmänkin oppia lisää ohjelman virtauksesta ja olisiko käyttökokemus ollut parempi kuin pelkissä asennusohjeissa.

Eräänlaisena vasta-argumenttina puhtaalle markkinoiden (loppukäyttäjien) ymmärtämisen kautta tapahtuvalle kehitystyölle on jättiyhtiö Applen edesmenneen toimitusjohtajan Steve Jobsin näkemys siitä, että joskus ihmisille pitää vain kertoa, mitä he haluavat. Tai mitä autoteollisuuden pioneerin Henry Fordin väitettään sanoneen: ”jos olisin kysynyt ihmisiltä mitä he haluavat, olisi vastaus ollut nopeampia hevosia”. Haasteena on siis vastuunottaminen käyttäjien näkökulmista ja siitä, mitä he oikeasti haluavat. Käyttäjätarinoita voidaan käyttää tunnistamaan käyttäjien pohjimmaisia tarpeita ja motivaatioita liittyen tuotteen tai palvelun käyttöön, mutta tietynlainen kriittinen tilannetaju on hyvä pitää silti takalalla mielessä.

MVP:hen liittyy kuitenkin oppiminen vahvasti, mikä tarkoittaa sitä, että pienimmän toimivan tuotteen toimittamisen jälkeen aletaan suuntaamaan jatkokehitykseen käyttäjäpalautteen tai muiden havaintojen pohjalta.

4 Toteutus

Tässä luvussa käydään läpi insinööriyön työn edistymistä jälkeinpäin tarkasteltuna sekä työn päätavoite toimitetaan alaluvussa 4.4.

4.1 Työmenetelmien tarkastelu

Lähtötilanne projektille oli otollinen. Saatavilla oli paljon hyvää lähdemateriaalia palvelimen hallinnointiin ja ohjelmistoihin liittyen. Loppujen lopuksi voitaneen sanoa, että työ oli varsin sopivasti mitoitettu tietoteknisten haasteittensa osalta.

Työtä tuki myös sen ohjaajat tarvittaessa, jolloin saatiin muun muassa tietoteknillisistä kommentteista lisää varmuutta omaan tekemiseen.

Työssä tehtiin aluksi minimaalinen toimiva versio (MVP), eli asennettiin webERP-palvelin toimintakuntoon keinoja kaihtamatta. Teorian osalta muodostettiin yleistä käsitystä siitä, mitä suurin piirtein tulisi tapahtumaan ja sitten alettiin iteroimaan lyhyiden teoriapätkien ja asennusohjeiden parantamisen välillä. Teorian lukeminen toi uusia näkökulmia ja toi esiin tapoja parantaa lähestymistä.

Ensimmäisten onnistumisten kautta tuli myös idea muuttaa lähtökohtaa asentamista kohtaan. Lisätavoitteena haluttiin tavoitella automatisoidun asennustyökälun rakentamista. Sen avulla olisi laajemmissa visioissa voinut, asentamisen automatisoinnin lisäksi, vaikuttaa webERP:n hallinnointiin. Ideointi oli luontaisesti edellä toteutusta ja työlista oli ruuhkautunut erilaisista ominaisuuksista.

Samalla saatiin ideoita tutkia omia työmenetelmiä itsessään ja mahdollisesti löytää väyliä tehokkaampaan työskentelyyn. Päätös tarkastella erilaisia malleja

pohjautuu muun muassa internethakujen tuloksiin ja ohjelmointifoorumien keskusteluihin, joissa oli koitettu ratkaista samanlaisia ongelmia, joita tämän insinööriyön aikana ilmeni.

4.2 Ketterät menetelmät

Insinööriyössä haasteena oli itsensä johtaminen asioiden oppimisen ja toisaalta tavoitteiden saavuttamisen välillä. Minimaalinen panostaminen tavoitteen saavuttamiseksi on eräs kultainen linjaus, mitä ei pidetä ketterän kehityksen tapauksessa laiskuutena vaan työtehona.

Paremmiin sanottuna, kyse on minimaalisen, mutta riittävän hyvän ratkaisun toimitamisesta. Teknistä rimaa nostetaan sitten laatuvaatimuksien kautta, jos halutaan.

Insinööriyössä eksyttiin välillä tekemään asioita ilman suunnitelmaa. Tämä johti joskus siihen, että keskityttiin epäoleelliseen laatuun. Vertauskuvallisesti, keskityttiin hienojen lattialaattojen valintaan, muttei vielä tiedetty saumaamisesta mitään.

Ketterien menetelmien kanssa puhutaankin usein oppimisesta näkökulmana työhön. Asiantuntijalla on jo näkemystä mihin suunnata, mutta ensikertalaiselle oppikokemus on välttämätön paha.

Ketterien menetelmien lyhyet toimitusvälit auttavat myös projektia pysymään raiteillaan, kun kokeilujen jälkeen pysähdytään arvioimaan tulosten vaikutusta. Teorian opiskelua on toki haasteellista johtaa, kun kerran asiantuntijuutta ei vielä ole, mutta ainakin tätä ajatusmaailmaa pyrittiin soveltamaan niin, ettei yliopiskeltaisi jotain tiettyä asiaa, vaan pyritään ensin palvelemaan minimaalista toimivaa ratkaisua.

Liiallinen oppiminen tietystä aiheesta syö aikaa pois muualta. Etsiessä ratkaisuja työssä ilmenneisiin ongelmiin, tuntui joskus tuskaiselta mennä muiden ai-

heiden pariin. Oppimiskokemusta olisi tehnyt joissain tapauksissa mieli syventää, mutta se taas lisäisi kapea-alaista ylioppimisen riskiä, josta oma luonne muutenkin kärsii. Jos projektin eteneminen jää odottamaan ylioppimisen tai muun esteen takia, arvon toimittaminen hidastuu. Tämä oli tärkeä työskentelyn näkökulma.

Kanban

Kanbanin visualisointiin taulujen kautta ei päästy syvemmin sisälle. Liian yksinkertainen taulu ei toisi paljoa arvoa ja monimutkainen tekisi informaatioähkyä. Visuaalisesta työn tilanteen hahmottamisesta on kyllä potentiaalinen hyöty työmuistin palauttamisessa, mikä voisi auttaa palaamaan nopeammin siihen, mihin viimeksi jäi työn parissa.

Visualisoinnin tarve tai oleellisuus lienee kontekstisidonnaista. Esimerkiksi voitaneen todeta, että työvaiheen tekijää ei tarvinne ilmaista yhden henkilön työstämässä projektissa.

Työssä eräs merkittävä ongelma oli juuri siinä, että siinä missä henkilö voi ottaa sahan käteen ja sahata lautaa tuosta noin vain, piti insinööriyössä muistutella itseä siitä mitä viimeksi tuli tehtyä. Ohjelmoidessa piti huomioida ohjelman koodin logiikka ja pystyttää työympäristö. Kirjoitustyössä piti muistella mitä kirjoitettavasta aiheesta asiasta on jo tullut mainittua. Otsikkorakenne auttaa hallitsemaan aiheita, joskin rakenne muuttui kirjoitustyön aikana.

On ilmeistä, että tehokkaamman työnteon kannalta olisi syytä ylläpitää työmuistia omassa mielessä tehden työtä jatkuvasti lyhyin väliajoin pois päiväjärjestyksestä. Työhön palaaminen maksaa hukattua aikaa. Asiaa visualisoidaan kuvassa 7, jossa ilmennetään tehokkaan työ ajan menetysten määrää, joka kasvaa aina sitä mukaan, mitä enemmän on samanaikaista työtä käynnissä.



Kuva 7. Työtehon menetyksistä, kun tehdään useampaa projektia samanaikaisesti. Kuvaaja tulee pitää enemmän viitteellisinä arvoina kuin kirjaimellisina (kuvan lähde: Hammarberg & Sunden 2014: alaluku 5.2.1).

Kanbanista kenties hyödyllisimpänä ajatusmallina jäi työn virtaamisen ajattelemisen, missä etenkin työn rajoittaminen on tärkeää. Samanaikaisesti käynnissä olevien työvelvoitteiden määrän rajoittaminen tehostaa työn tekemistä ehkäisten muun muassa edellä mainittuja "vaihtotappioita". Arvon luomisen kannaltakin on parempi saada yksi projekti valmiiksi kuin tehdä neljää samanaikaisesti odotellen toimituksia.

Personal Extreme Programming

PXP:llä on hyvä myyntipuhe ja teoriassa sopisi erittäin hyvin insinööriyön tilanteeseen, jonka erityispiirteenä oli yhden hengen ryhmän ohjelmointityö. PXP:n tarjoama malli on kuitenkin melko tekninen ja vaatii asiantuntemusta.

Käyttäjätarinoiden laadintaan oli muodostunut opintojen kautta hieman kokemusta ja tätä käytäntöä käytettiin jonkin verran. Pyrittiin siis asennoitumaan henkilön asetelmaan, joka ei ole tutustunut esimerkiksi palvelimien toimintaan.

Käyttäjätarinan näkökulmaa pyrittiin hyötykäyttämään sekä asennustyökalun kehityksessä, että ohjeiden laadinnassa. Ohjeiden laadinta pyrittiin toteuttamaan näkökulmasta, jota osaisi seurata ihminen, jolla on tietokoneiden käytöstä kokemusta, muttei mitään erityistä asiantuntijuutta.

4.3 Näkemyksiä palvelimen valintaan

4.3.1 webERP:n vaatimusten täyttäminen

WebERP:tä esiteltiin luvussa 3.2. Sen omasta ohjekirjasta löydettiin maininta järjestelmän vaatimuksista, jotka jaettiin kahteen osa-alueeseen: laitteisto- ja ohjelmistovaatimuksiin. Siellä ei kuitenkaan otettu tarkemmin kantaa, toisin kuin joidenkin kaupallisten ERP-järjestelmien kohdalla, miltä suorituskyvyn vaatimukset näyttäisivät numeroina, esimerkiksi RAM-muistin osalta.

Laitevaatimuksissa avainhavainto oli järjestelmän keveys, joten pienellä kuormituksella ei todennäköisesti tarvitsisi tilata pientä kokoluokkaa suurempaa palvelinta Educloudin palvelukatalogista. Toisaalta webERP:n ohjeissa, kuten yleensäkin, mainitaan, että isompi käyttäjämäärä palvelimella tarkoittaa, että palvelinta tulisi skaalata vastaavasti tehokkaammaksi suorituskyvyltään, jotta se selviää palvelupyynnöistään.

Palvelimien laitevaatimuksia kartoitettiin yleisesti eikä mitään yleispätevää sääntöä ollut helppo muodostaa. Jotkin kaupalliset ohjelmistot olivat raskaita ja vaativat selvästi enemmän prosessointitehoa kuin toiset.

Lisäksi Metropolian tietohallinto pyytää palvelimen tilaajaa mitoittamaan tarpeensa oikein, jottei ylimääräisiä resursseja tarvitse rajata pois muilta mahdollisilta palvelin palvelun käyttäjiltä (Keski-Ruismäki 2018).

Tarkoista luvuista ei löydetty viitteitä, mutta vertailuna löydettiin eri nettilähteistä kaupallisten toiminnanohjausjärjestelmien laitteistosuosituksia, joissa laitteiston

tehovaatimukset tai -suositukset olivat joissain tapauksissa minimivaatimuksiensa osalta jopa enemmän kuin mitä Educloudin katalogista voi tilata enimmillään.

Onneksi webERP onkin kehittäjänsä mukaan ”poikkeuksellisen kevyt” järjestelmä. Ohjekirjan mukaan sitäkin voi skaalata ylös, etenkin lisäämällä palvelimen RAM-muistin määrää. RAM-muistin lisääminen mainittiin myös muissa lähteissä ja sitä tarvitaan muun muassa käyttäjien (samanaikaisten palvelupyyntöjen) määrän kasvaessa. Ohjekirjassa mainittiin, että ”lähtötason” palvelin voisi palvella viittäkymmentä samanaikaista käyttäjää. Lähtötasoa tosin ei määritetty, ja internethakujen perusteellakin selvää näkemystä 2000-luvun alun lähtötason palvelimesta ei pystytty ollenkaan tarkasti arvioida. Asian osalta syvällisempään selvittelytyöhön ei päätetty panostaa.

Testikäytössä webERP:tä kokeiltiin Educloudin kevyimmällä palvelimella ja järjestelmä pyöri ilman havaittavia ongelmia, joskin ainoa kuormitus yksittäisen käyttäjän toimesta on sivujen lataamisessa ja datan rivikohtaisissa tietokantatalennuksissa.

Käytännössä voisi tehdä niin, että jos palvelimen toiminta alkaisi hidastua useammalla käyttäjällä käytettäessä, voisi Educloudista tilata suuremman palvelimen. Insinööriyön asennusohjeissa kallistuttiin keskikokoisen palvelimen suositamiseen.

Liian tehokkaan palvelimen tilaaminenkaan useaksi kuukaudeksi ei ole sekään peruuttamaton virhe, sillä palvelimen elinkaaren voi lopettaa myös ennen eräpäivää. Tästä lyhyet ohjeet alaluvussa 4.4.1, jossa esitellään palvelimen tilaamista Educloud-palvelusta.

Tietokantajärjestelmän käyttäjätillillä, jonka kautta webERP:tä käytetään, tulee olla oikeudet luoda tietokantoja ja tauluja. Ohjekirjaa lukemalla muodostui käsitys, että asennusohjelma testaa tarvittavat vaatimukset itse ja ilmoittaa virheilmoituksella käyttäjälle, jos asennusta ei voida jatkaa. Myös käyttöoikeudet tulisi olla olemassa nettipalvelimen asiaan kuuluviin kansioihin.

Palvelimella oli eri ohjelmistoja, joiden oli tarkoitus toimia yhteisellä keskenään. Versionhallinta muodostui työssä erääksi haasteeksi, kun ohjelmien välillä oli ristiriitaisuuksia ja siksi päätettiin asentaa spesifit versiot PHP:stä sekä MariaDB:stä, jolloin ei olla niin epävarmassa asemassa eriävään tahtiin päivittyvien ohjelmistojen kanssa.

PHP:n ja webERP:n välillä havaittiin ongelma, sillä webERP:n silloiseen versioon oli kirjoitettu liian vanhaa PHP-koodia. Tarkemmin sanottuna, lähdekoodissa oli funktio, joka oli poistettu uudemmassa PHP:n versiossa.

Ratkaisuksi opiskeltiin, kuinka palvelimelle annetaan käskyt tiettyjen ohjelma-versioiden asentamisesta ja sitten tutkittiin mitkä, mieluummin mahdollisimman uudet, versiot sopivat yhteen keskenään. Ohjelmien versioiden hallinnasta mainitaan vielä luvussa 5.

4.3.2 Käyttöjärjestelmän käyttökokemus

Insinööriyön ensimetreillä pohdittiin, mikäli Windows-järjestelmä olisi käyttäjälle helpompi ja tutumpi käyttää Linux-pohjaiseen järjestelmään verrattuna. Tavanomaiseen tietokoneen käyttämiseen verrattuna kulttuurishokki on joka tapauksessa ilmeinen graafisen käyttöliittymän puuttuessa.

Palvelimen ylläpitäjän näkökulmasta hakemistorakenne on erilainen Windowsin ja Linuxin välillä: Windows-koneissa tiedostot ovat jaettu erinimisiin asemiin, joita kuvataan yhdellä kirjaimella, kun Linuxissa hakemistorakenne pohjaa yhteen juurikansioon, jossa järjestys luodaan laittamalla tietyn tyyppiset tiedostot saman hakemiston alle. Koski (2010: 72) mainitsi, että Linuxin tiedostorakenteessa yleensä löytää nopeasti kaipaamansa tiedostot, kun järjestyksen logiikan on oppinut, mikä toi luottoa Linuxin valintaan, vaikkei kokemusta insinööriyöhön lähdetäessä itsellä ollut Linux-järjestelmistä.

Internethauilla sekä kirjallisuudesta löydettiin paljon ohjeistusta Linuxin käyttöön, joten ainakaan lähteiden puuttuminen ongelmanratkaisun osalta ei aiheuttanut huolta. Linux on muutenkin suosittu vaihtoehto ammatillisissa asiayhteyksissä.

4.3.3 Järjestelmän tehokkuus

Kenties pienellä varauksella yleistettynä; Linux on tehokkaampi käyttämään tietokoneen resursseja kuin Windows. Tämä heijastunee siinäkin, että Windows-palvelimelle suositus oli valita vähintään keskikokoinen kokoonpano, siinä missä Linuxille sopi myös pieni (Keski-Ruismäki 2018).

Resurssien hyödyntämisen puolesta valinta kallistui Linux-pohjaiseen järjestelmään. On vaikea arvioida, onko lievällä säästöllä käyttötärpeeseen suhteessa juurikaan eroa, kun itse webERP on kevyt. Oletettavasti eroja muodostuisi tilanteissa, mikäli palvelimelle kohdistuisi useita eri pyyntöjä samanaikaisesti.

WebERP:n palvelin toimii PHP-kielellä, joka suorittaa palvelimeen kohdistuvia pyyntöjä palvelimella, kun verrataan esimerkiksi JavaScriptiin, joka ohjaa prosessointia palvelun käyttäjän koneelle. Prosessointivastuuta ei voi tässä tapauksessa helpottaa, koska se vaatisi webERP:n vaihtamista toiseen koodikieleen pois PHP:stä. Joskin tämä ei välttämättä aiheuta ongelmaakaan, kun webERP on suunniteltu kevyeksi järjestelmäksi toiminnaltaan. Mikäli opetustilanteissa muodostuisi jatkuvaksi ongelmaksi palvelimen hidas vaste, kun siihen kohdistuisi monta samanaikaista pyyntöä, olisi ainoa helppo ratkaisu vain tilata uusi palvelin kokoluokkaa tehokkaammalla kokoonpanolla.

4.3.4 Elinkaarisuunnittelu

Palvelimen tarjoaja tarjoaa palvelimen maksimissaan muutamaksi kuukaudeksi ja tämän jälkeen sen käyttöä voi vielä jatkaa parilla viikolla pyydettyä. Lisäksi erityistapaukset lienevät neuvoteltavissa erikseen. Aikaraameissa pysyessä ei tarvitse tehdä mitään erityisjärjestelyitä palveluntarjoajan kanssa. Palvelimia voi tilata useammankin samanaikaisesti.

Konteksti opetustilanteisiin on pikaisesti arvioituna helppo hallinnoida elinkaarensa kannalta. Kevyellä järjestelmällä tuskin tulee skaalaustarvetta, kun oletettavasti samanaikaisten käyttäjien määrät jäänevät korkeintaan muutamaan kymmeneen. Yrityskontekstit lienevät monimutkikkaampia, kun elinkaaret suunniteltaisiin oletettavasti useiden vuosien mittaisiksi huomioiden esimerkiksi ajan kanssa kasvava käyttäjien ja datan määrä.

4.4 Asennusohjeet

Tässä alaluvussa käydään opastaen läpi insinööriyön päätavoitteena ollut webERP-toiminnanohjausjärjestelmän asentaminen. Alla taulukko 1 esittää koko asennusprosessin tiivistetysti. Asennuskoodista luotiin myös tiivis versio työn loppuun liitteeksi, jonka voi kopioida ja liittää suoraan palvelimelle asentamisen nopeuttamiseksi. Liite ajaa samat käskyt, jotka esitellään alaluvussa 4.4.3.

Järjestelmän ylläpitoon ja päivittämiseen ei oteta kantaa. Luomme kuitenkin hallinnointia varten tilit alaluvussa 4.4.2. Ohjeet alkavat itse palvelimen tilaamisella Educloud-palvelusta alaluvussa 4.4.1.

Taulukko 1. Asennusprosessi webERP-palvelimelle.

Vaatimukset/lisätiedot	Suorite	Tuotos
<ul style="list-style-type: none"> Resurssitarpeiden/elinkaaren pohtiminen Metropolian tietohallinnon dokumentaatio Yhteys Metropolian verkon kautta (tarvittaessa VPN) 	Palvelimen tilaaminen	<ul style="list-style-type: none"> LAMP-pinollinen palvelin tilattu Palvelimen tiedot toimitetaan sähköpostiin
<ul style="list-style-type: none"> Palvelimen tiedot (IP-osoite ja salasana) VPN-yhteys tarvittaessa phpMyAdmin:lla voi tutkia graafisella käyttöliittymällä palvelimen tietoja 	Muodostetaan yhteys palvelimeen	<ul style="list-style-type: none"> Yhteys palvelimeen kunnossa
<ul style="list-style-type: none"> Hyvänä käytänteenä luodaan hallinnointitilit Vaiheen voi ohittaa tai palata siihen jälkikäteen 	Käyttäjätilien luominen (vapaaehtoinen)	<ul style="list-style-type: none"> Käyttöjärjestelmän hallinnoija (CentOS) Tietokannan hallinnoija (MariaDB)
<ul style="list-style-type: none"> Käyttöjärjestelmään asennetaan sekä päivitetään ohjelmia webERP:n tiedostot ladataan netistä 	Tarvittavien ohjelmien ja webERP:n asentaminen sekä päivittäminen	<ul style="list-style-type: none"> Tukevat ohjelmistot asennettu webERP valmis selainpohjaiseen asennusohjelmaan
<ul style="list-style-type: none"> Käytettävä tietokantatili on valittu (juurikäyttäjä, hallinnoija tai jokin muu) 	webERP:n asennusohjelman ajaminen	<ul style="list-style-type: none"> Admin-tunnus luotu asennusohjelmassa salasanan kera webERP asennettu

Taulukkoa 1 voi käyttää viitteenä tai opasteena asennusta tehdessä. Tuotos-sarake ilmaisee kunkin vaiheen (suoritteen) tärkeimmät tavoitteet.

4.4.1 Palvelimen tilaaminen

Palvelimen vaatimuksia käytiin läpi aiemmin alaluvusta 4.3 alkaen, jolloin määritettiin mitä oltiin tilaamassa. Seuraavaksi käydään läpi, miten palvelin tilataan Metropolian Educloud-palvelukatalogista.

Educloud

Educloud-palvelun toimittaa Metropolian tietohallinto. Tähän projektiin liittyvän ympäristön rakentaminen alkaa Metropolian tietohallinnon ”opetuskäytön educloud virtuaalipalvelimet” -sivuilta (wiki.metropolia.fi/pages/viewpage.action?pagelId=132852058), jossa on ohjeet virtuaalipalvelimen tilaamisesta. Tämä osa prosessia on siis dokumentoitu tietohallinnon puolesta ja alla näytetään, kuinka se toteutuu käytännössä.

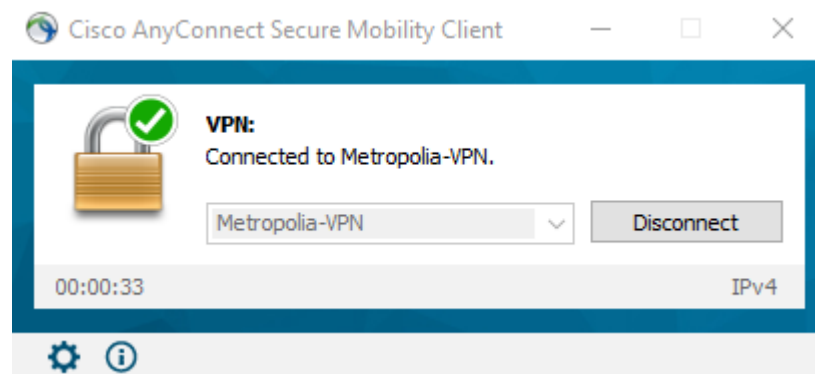
Käyttäjän tulee ensiksi mennä Educloudiin, osoitteeseen: educloud.metropolia.fi/. Internet-yhteys tulee olla Metropolian verkon kautta. Tämän voi todentaa menemällä Educloudiin ja tarkastelemalla, latautuuko sen sivusto vai ei. Jos sivu latautui, käyttäjä on Metropolian verkossa ja jos ei, tarvitaan joko VPN-yhteys verkkoon tai sitten tilaaminen (ja palvelimen käyttäminen) tulee suorittaa paikan päältä Metropolian toimipisteellä olevalta koneelta. VPN-ohjeita, joita käsitellään seuraavan otsikon alla, ei tarvitse lukea, jos käyttäjä pääsee Educloud-sivulle ja kirjautumaan sisään tunnuksillaan.

VPN-yhteys

VPN-yhteys muuntaa käytetyn laitteen IP-osoitteen Metropolian verkon kautta toimivaksi. Tämä osa prosessia on dokumentoitu tietohallinnon ”VPN-etäyhteydet”-sivulle (wiki.metropolia.fi/pages/viewpage.action?pagelId=149652071).

Huomionarvoista on, että tietohallinto pyytää käyttämään VPN-väylää vain asianmukaiseen internetliikenteeseen, koska palvelun käyttö varaa Metropolian omaa tietoliikennekapasiteettia. Eli, kun VPN-yhteys on luotu, ei käytetä netti-kaistaa vaikka viihdekäyttöön, sillä kapasiteetti on muualta pois.

Insinööriyössä käytettiin tietohallinnon dokumentaatiossa mainittua AnyConnect-ohjelmaa (ohjelman näkymä kuvassa 8), sillä työ tehtiin etänä omasta kotoa käsin (verraten valmiiksi Metropolian verkossa olevalta koneelta). AnyConnect ladattiin omalle koneelle, mutta se toimii myös selaimen kautta ilman asennusta.

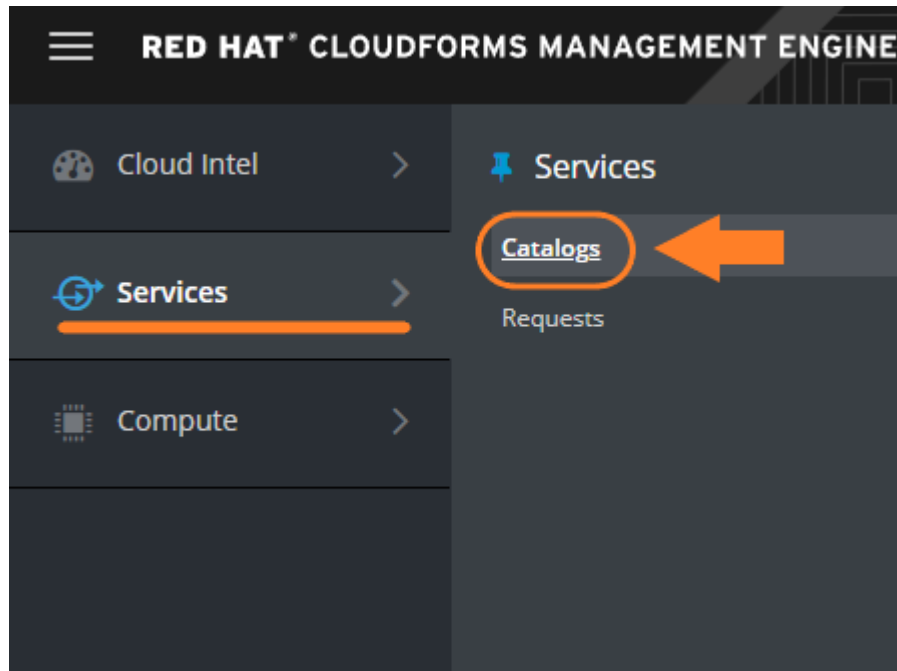


Kuva 8. AnyConnect on muodostanut VPN-yhteyden Metropolian verkkoon.

AnyConnectin käytöstä puhutaan tietohallinnon aiemmin mainitussa VPN-dokumentaatiossa aliotsikossa "VPN-apuohjelman asennus ja käyttö omissa tietokoneissa". Kuvan 8 näkymässä yhteys on muodostettu onnistuneesti Metropolian verkkoon.

Palvelimen valinta

Kuten aiemmin päätettiin, webERP:tä varten tilataan Linuxin omaava palvelin ja valitaan palvelu, jossa on LAMP-pino esiasennettuna. Educloudissa valitaan kuvan 9 mukaisesti "Services → Catalogs".



Kuva 9. Näkymä Educloudin valikoista.

Vaihtoehtoisesti saattaa riittää, jos käyttäjä klikkaa pelkästään "Services":n kohdalta. Lopulta pitäisi päästä kataloginäkömään, jossa lukee "All Services" kuvan 10 mukaisesti.

All Services

	Name	Description	Tenant	
	Centos 7 x64	Basic Install with IP address setup done	My Company	Order
	<u>Centos 7 x64 LAMP</u>	Centos 7 with Apache, MariaDB, PHP installed	My Company	Order
	Centos 7 x64 Self install	Install Centos 7 x64 from scratch	My Company	Order
	Windows 2008R2 Self Install	Self install Windows 2008R2 from scratch	My Company	Order
	Windows 2012R2	Basic Install with IP address setup done	My Company	Order
	Windows 2012R2 Self Install	Self install Windows 2012R2 from scratch	My Company	Order
	Windows 2016	Basic Install with IP address setup done	My Company	Order

Kuva 10. Katalogista valitaan valmis LAMP-pino.

Tavoite oli asentaa LAMP-pinollinen palvelin. Käyttäjä valitsee kuvan 10 mukaisesti "Centos 7 x64 LAMP":n riviltä sinisen painikkeen, jossa lukee "Order".

Seuraavaksi jätetään tilauspyyntö. Kuvassa 11 käyttäjä valitsee palvelimen koon ja käyttöajan.

WebERP:n ollessa itsessään poikkeuksellisen kevyt järjestelmä, voitaneen mahdollisesti valita jopa pienikokoisin palvelin, valitsemalla "S - Small", mutta lienee perusteltua pelata varman päälle ja tilata keskikokoinen koko "M - Medium". Nyrkkisääntö webERP:n ohjeissa oli, että kymmeniä samanaikaisia käyttäjiä varten riittäisi "lähtötason" palvelin, miten tämän sitten osaa tulkita. Yhdellä käyttäjällä järjestelmään riittää pienin kokoonpano.

Palvelimen käyttöajan voi valita toki tarpeen mukaan. Pisin aika on neljä kuukautta ("4 months") ja siihen voi vielä anoa lisääaikaa päälle. Tästä mainitaan tilauksen yhteydessä saatavassa automaattisessa vahvistussähköpostiviestissä.

Lopuksi vielä hyväksytään käyttöehdot ruksaamalla ruutu kohdassa "Accept terms of use" ja vahvistetaan tilauksen jättäminen painamalla alhaalla oikealla sijaitsevaa sinistä painiketta "Submit" (katso kuva 11).

Order Service "Centos 7 x64 LAMP"

Provisioning Information

Size S - Small M - Medium L - Large

Lease time 1 day 1 week 1 month 2 months 3 months 4 months

Terms of use

General terms of use for EduCloud.metropolia.fi and it's VM's

* Educloud Virtual machines are ment to be used only for acedemic study purposes for metropolia's courses and projects.

* You are not allowed to run any HPC applications, Bitcoin miners or any other distributed processing applications on educloud.

* IT Services reserves right to Poweroff / Reset / Delete any VM that is

Accept terms of use ←

➔

Kuva 11. Valitaan palvelimen koko ja laina-aika.

Palvelimen tiedot

Kun olemme oikeassa verkossa (tarvittaessa VPN:n kautta) tarvitsemme vielä palvelimen IP-osoitteen ja sen käyttöjärjestelmän juurikäyttäjän salasanan. Tietohallinto luo molemmat edellä mainitut palvelimen tilaajan puolesta ja lähettää tiedot niistä sähköpostilla tilaajalle.

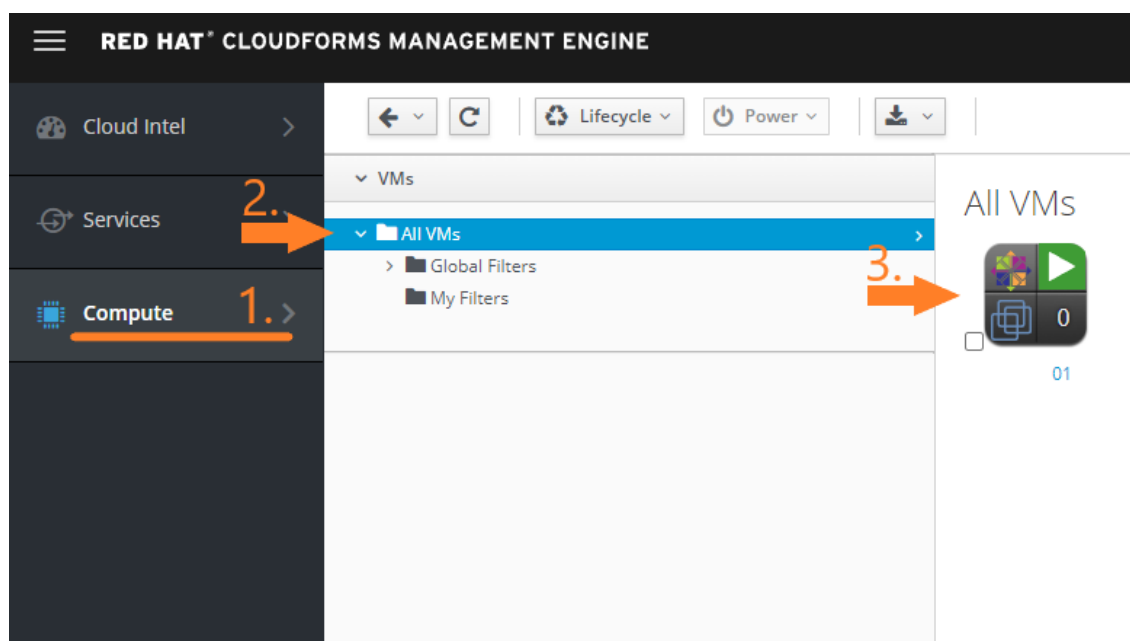
Viesti omaa hieman hämäävästi montakin erilaista IP-numeron näköistä elementtiä, mutta kaipaamamme IP-osoite lukee rivillä, joka on muotoa "IP: <IP-osoite>" ja salasana lukee muutaman rivin alempana rivillä "Password: <salasana>".

Palvelimen elinkaaren päätös

Kuvassa 11 nähtiin "Lease time", jossa valitaan hankintahetkellä palvelimen käyttöaika. Lisääaikaa voi tosin anoa Educloudissa (sekä oletettavasti helpdeskin kautta) alkuperäisen laina-ajan jatkamiseksi.

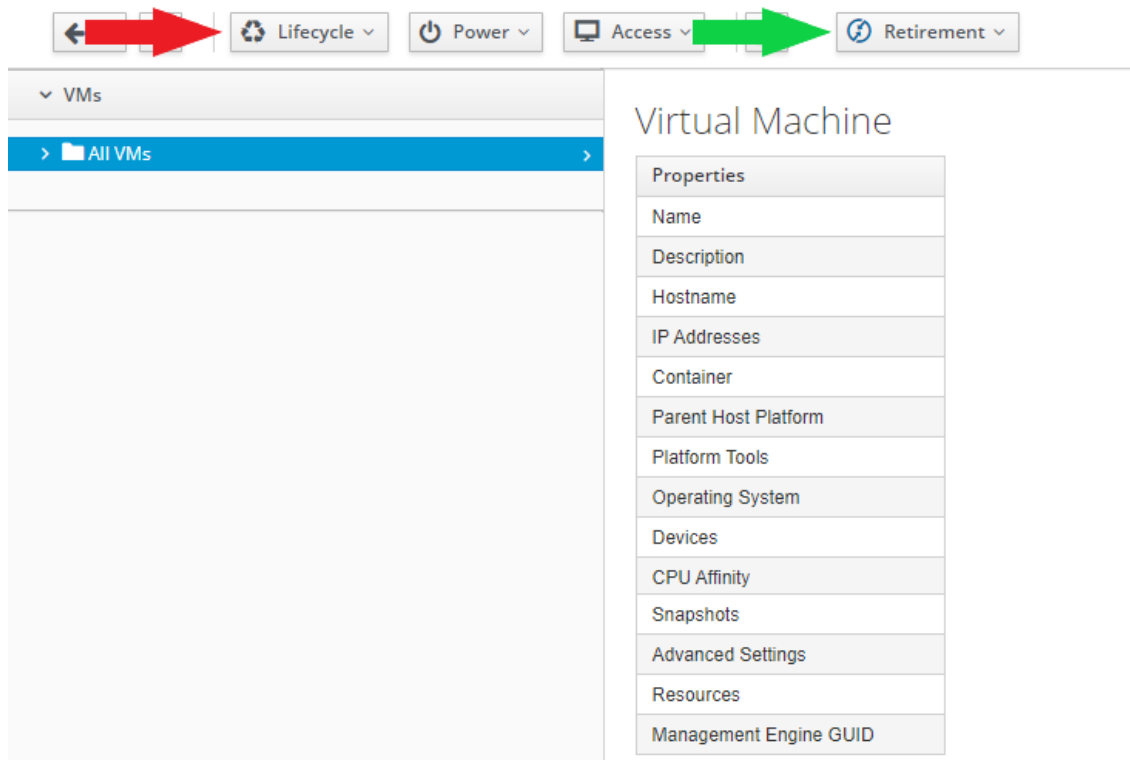
Elinkaaren voi myös lopettaakin ennenaikaisesti. Tämä voisi olla hyödyllistä tilanteissa, joissa tiedetään, ettei pitkää lainaa tarvitsekaan jatkaa, tai joissa on tullut hankittua pitkäksi aikaa liian tehokas palvelin ja täten voidaan vapauttaa resursseja takaisin Metropolialle.

Palvelimen tilauksen elinkaaren muutos tapahtuu kuvan 12 mukaisesti Educloudissa valitsemalla "Compute" vasemmanpuoleisesta päävalikosta ja sen jälkeen valiten oikealta palvelin, johon halutaan vaikuttaa. Jos palvelinta tai palvelimia ei näy oikealla (kuvassa 3. kohta), vaikka pitäisi, tulee varmistaa, että kohdan 2 mukaisesti on "All VMs" valittu rakennevalikosta.



Kuva 12. Palvelimen löytäminen "VMs"-valikon kautta.

Palvelimen kuvakkeen klikkaamisen jälkeen saavutaan sen omaan valikkoon. Kuvassa 13 valitun palvelimen tiedot näkyvät oikealla taulukkona. Tässä näkyvässä kyseessä oleva palvelin voidaan joko luovuttaa pois käytöstä valitsemalla punaisen nuolen osoittama "Lifecycle"-painike, tai anoa sille lisääaikaa vihreän nuolen osoittamasta "Retirement"-painikkeesta.



Kuva 13. Valitun palvelimen elinkaaren voi päättää Lifecycle-painikkeella tai anoa jatkoa Retirement-painikkaan kautta.

Termi "retirement" johtaa helposti harhaan. Ylävalikon painikkeiden painaminen kertaalleen erheellisesti ei tee välittömästi mitään peruuttamatonta.

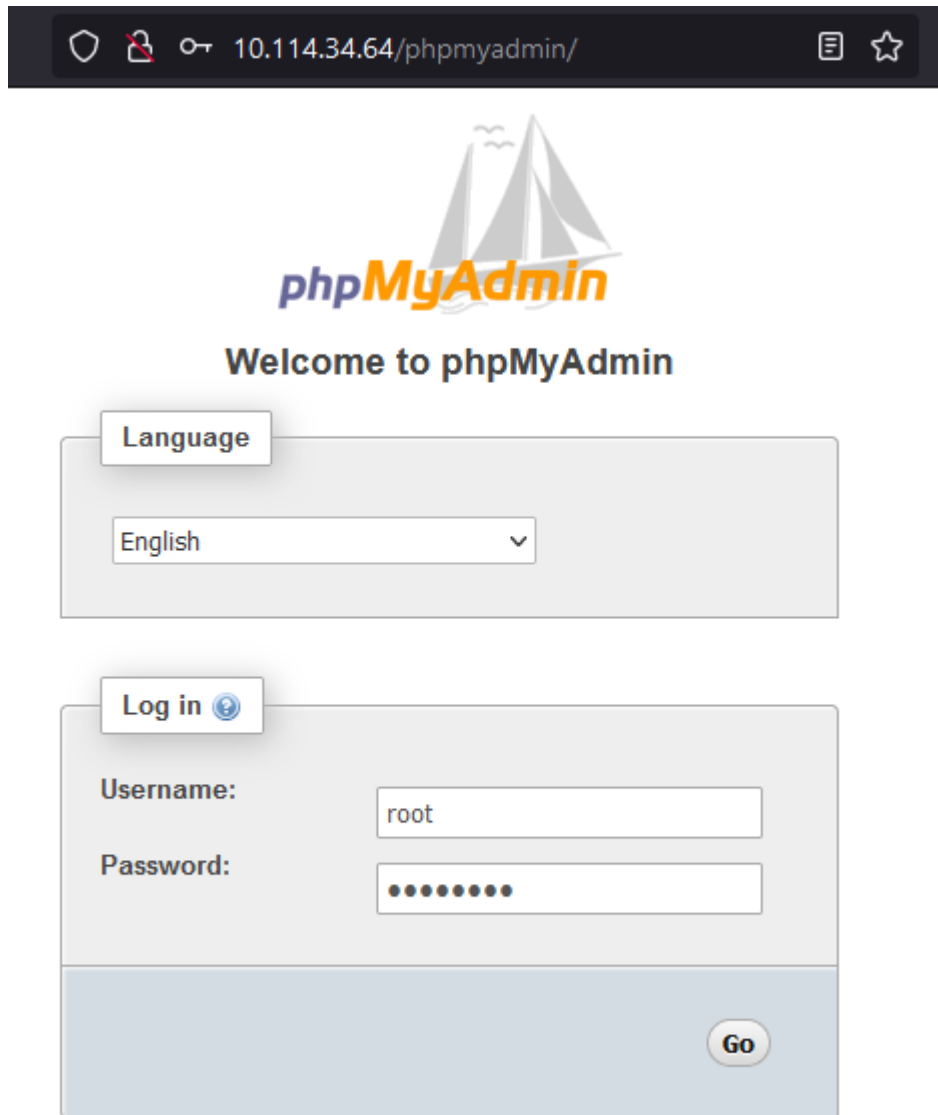
Yhteys palvelimeen selaimella

Tämän otsikon alla kokeillaan toimiiko yhteys nettipalvelimeemme. Jos yhteys ei toimi, tulee käyttäjän varmistaa, että yhteys Metropolian verkon kautta ja tarvittaessa palata aiempaan otsikkoon "VPN-yhteys". Kokeilu suoritetaan selaimen kautta menemällä nettipalvelimelle asennettuun phpMyAdmin-ohjelmaan,

johon tutustutaan ohimennen. Tämän vaiheen voi myös ohittaa ja siirtyä otsikkoon ”yhteys palvelimeen komentotulkilla”.

PhpMyAdmin-ohjelma toimii omanlaisena hallinnoijan graafisena käyttöliittymänä palvelimelle, erityisesti antaen väylän tietokantojen hallintaan. Alla alustetaan weberp-niminen tietokanta, mutta tämänkin voi ohittaa webERP:n asennusohjelman hoitaessa saman asian. Itse tehtynä etukäteen voinee tosin vaikuttaa tietokannan collation-arvoon, mutta se ei liene kovin oleellista.

Seuraavaksi muodostetaan yhteys phpMyAdmin-ohjelmaan, joka on tullut LAMP-pinon mukana. Jos kokeilu ei onnistu, ei webERP:n asentamistakaan voine edistää. Mennään (omavalintaisen) internetselaimen kautta URL:iin <IP-osoite>/phpmyadmin/. IP-osoitteen syntaksin sekä yleisen näkymän voi tarkistaa kuvan 14 esimerkkitaipauksesta. Jos sivu ei lataudu, tulee mahdolliseksi virheiksi mieleen väärinkirjoittaminen, tai se, ettei kone ole Metropolian verkossa ja yhteys pitäisi muodostaa VPN:n kautta (katso aiempi otsikko ”VPN-yhteys”), tai että palvelimen asennus on vielä kesken (jolloin odotetaan varttitunti) tai kenties valittiin väärän palvelimen asennus palvelukatalogista.



10.114.34.64/phpmyadmin/

phpMyAdmin

Welcome to phpMyAdmin

Language

English

Log in

Username: root

Password:

Go

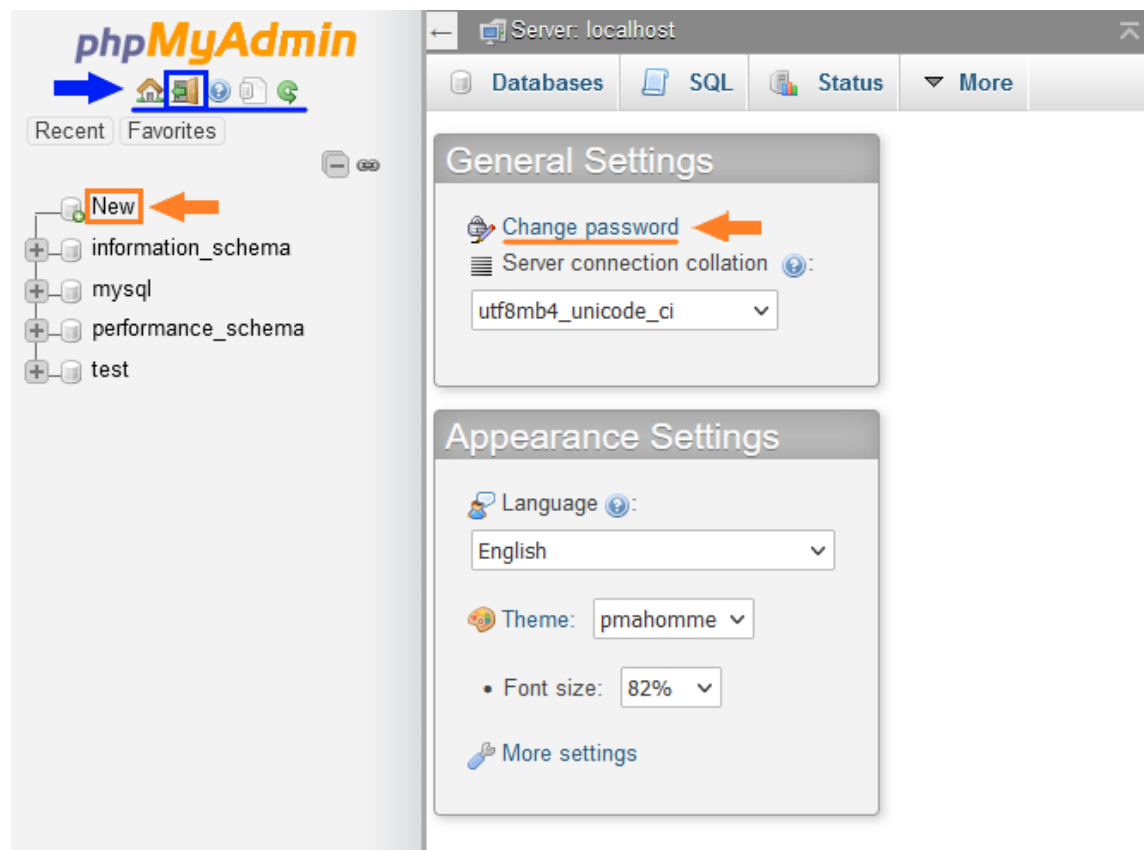
Kuva 14. Sisäänkirjautuminen selaimen kautta phpMyAdmin-ohjelmaan.

Käyttäjätunnukseksi ("Username") annetaan "root" ja salasanaksi ("Password") annetaan sähköpostina tulleisiin palvelimen tilaustietoihin saatu salasana.

PhpMyAdmin-sivuilla voidaan vaihtaa tietokantaohjelman juurikäyttäjän salasana. Kyseiseen palveluun kirjaudutaan nimenomaan tietokantaohjelman, eikä verraten palvelimen käyttöjärjestelmän, juurikäyttäjän salasanalla. Palvelimen käyttöjärjestelmän salasanan voisi myös halutessaan vaihtaa. Salasanan vaihtaminen on hyvä käytäntö, mutta toisaalta kun kontekstina on opiskelukäyttö,

jossa ei ole oikeaa dataa, ja jossa muutkin riskit ovat minimaalisia, ei tällä liene väliä.

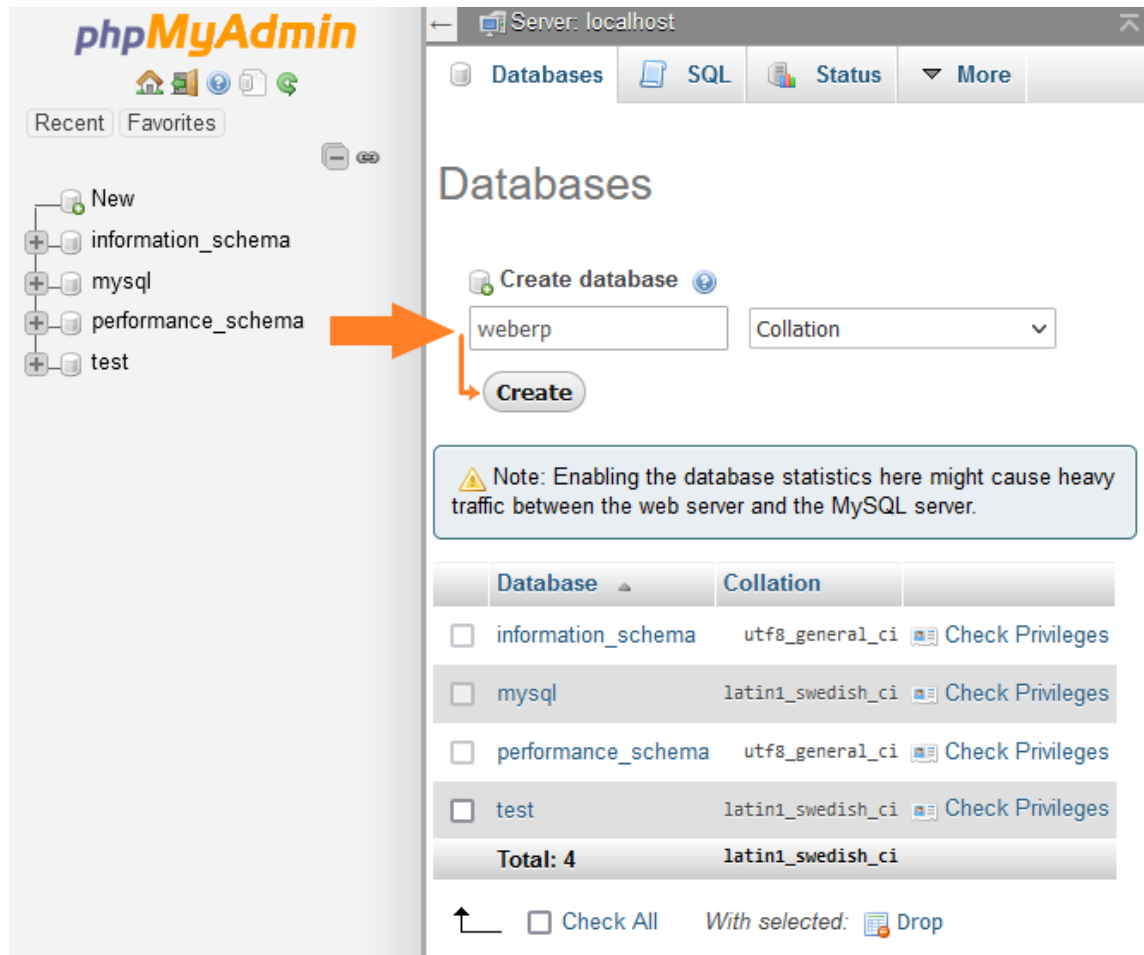
Kuvassa 15 on onnistunutta sisäänkirjautumista seurannut näkymä. Seuraavaksi alustetaan webERP:n asennusprosessia luomalla tietokanta sen tallettamiä tietoja varten. Tämän voi ohittaa, sillä webERP osaa luoda saman itsekin. Samassa kuvassa on sinisellä korostettu vasemmalla ylhäällä toinen ikoni vasemmalta, jota klikkaamalla voi lopuksi kirjautua ulos hallintaohjelmasta. Sitä ennen luodaan uusi tietokanta valitsemalla "New" näkymän vasemmasta laidasta.



Kuva 15. Näkymä phpMyAdmin-ohjelmasta. Sinisellä on huomioitu uloskirjautumispainike. Salasanan vaihtaminenkin on mahdollista.

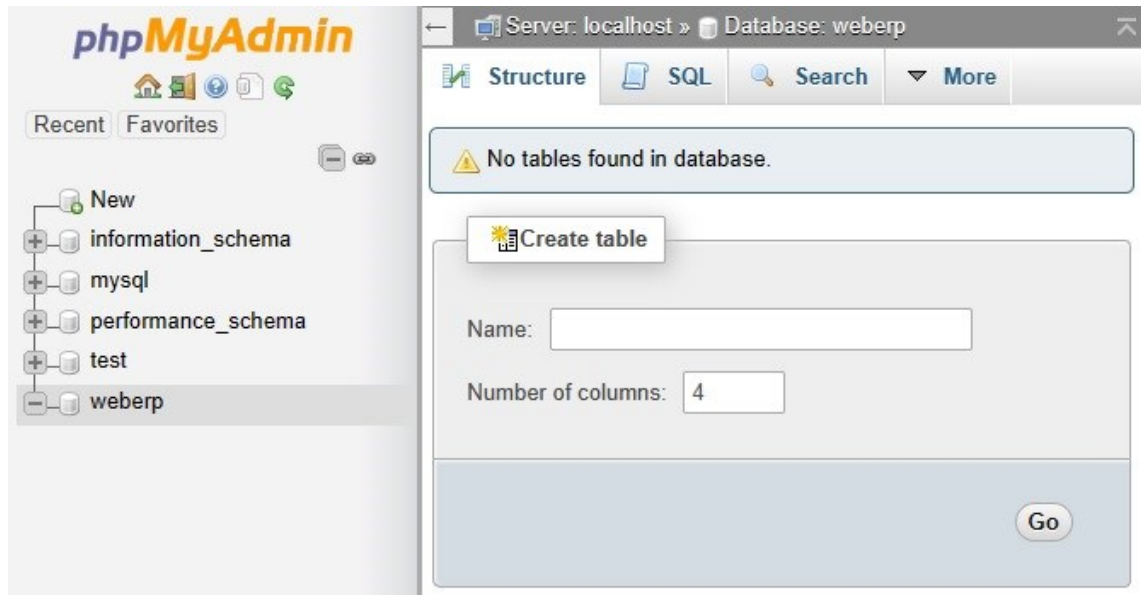
"New" valinnan jälkeen saavutaan näkymään (kuva 16), johon voidaan valita tietokannan nimi. Tässä annetaan nimeksi "weberp". Viereinen valikkoruutu, jossa lukee "Collation", liittyy merkistöön. Hobson ja Pelz (2016: Managing a MariaDB

database) suosittavat käyttämään UTF-8 -standardia, joten käyttäjä voisi valita collation-pudotusvalikosta "utf8_general_ci". Oletusarvona (ilman valintaa) asentui ainakin insinööriyön aikaan "latin1_swedish_ci", joka lienee myös kelvollinen.



Kuva 16. Tietokannan luontia graafisen käyttöliittymän kautta.

Painetaan "Create", kun nimi, ja mahdollisesti collation-arvo, on annettu. Saavutaan kuvan 17 mukaiseen näkymään. Tietokanta nimellä "weberp" on nyt luotu. Taulujen luonti jätetään webERP:n asennusohjelman harteille.



Kuva 17. Luotu tietokanta ilmestyi järjestelmän listaukseen vasemmalle.

PhpMyAdminista voi vielä halutessaan kirjautua ulos kuvan 15 mukaisesti. Painike on vasemman ylärivin ikoneista toinen vasemmalta.

Yhteys palvelimeen komentotulkilla

Tavoitteena on antaa käskyjä palvelimelle. Tietohallinto on dokumentoinut PuTTY:n käytön (Keski-Ruismäki 2016), mikä on eräs vaihtoehto. Insinööri-työssä käytettiin Git Bash:ia omalta Windows 10 -kotikoneelta. PuTTY koettiin hitaammaksi asettaa käyttövalmiuteen.

Tulkin valinta ei vaikuta koodin sisältöön. Tärkeää on huomata SSH-toiminnallisuus (tai vastaava), joka auttaa salaamaan tietoliikennettä. SSH löytyy PuTTY:sta ja Bashista.

Komentotulkki toimii yhdellä rivillä. Tulkillle kirjoitetaan komento ja se laitetaan toteutukseen painamalla Enteriä. Tulkin voi myös käskellä tulkitsemaan skriptitiedostoja, joissa on monta riviä koodia samassa nipussa.

Bashin asentamista ei ole dokumentoitu tähän, mutta ohjelman kotisivu löytyy osoitteesta git-scm.com/downloads.

Bashilla muodostetaan SSH-yhteys palvelimeen syntaksilla:

```
ssh <käyttäjä>@<IP-osoite> -p 22
```

Tässä tarvitaan siis kaksi syötettä. Käyttäjä on ensikertaa "root" ja jatkossa mahdollisesti jokin muu käyttäjätili, mikäli sellainen luodaan myöhemmin (hyvän käytänteen mukaisesti). IP-osoite on sama palvelimen IP-osoite kuin ennen. Lopussa oleva parametri "-p 22" ohjaa liikenteen portin 22 läpi, mikä ohjeistettiin tietohallinnon PuTTY-ohjeissa.

Tiedot täytetään edellä mainittuun syntaksiin ja komentotulkki kysynee ensikertaa kirjautuessa, halutaanko jatkaa. Tällöin komentotulkkiin voi kirjoittaa "yes" ja lähettää komento Enterillä.

Käytännön vinkkinä tekstin liittämiseen: kun Bashia käytetään Windows-koneella, ei pikanäppäinyhdistelmä Ctrl+V leikepöydän liittämiseksi toimi, vaan Bashin komentotulkkiin tekstiä liitetään Shift+Insert -yhdistelmällä tai hiiren oikealla painikkeella valikon kautta.

4.4.2 Käyttäjätilit

Hyvien käytänteiden mukaan juurikäyttäjätilejä (root) ei tulisi käyttää hallinnollisiin asioihin, joten tehdään palvelimen käyttöjärjestelmään (CentOS) ja tietokantaan (MariaDB) tilit hallinnointia varten.

Tämän otsikon voi periaatteessa ohittaa ja siirtyä seuraavaan. Ohittaessa asennus tapahtuu root-tilin ja sitä vastaavan salasanan kautta. Se on nopeuttaa asennusta hieman, eikä liene juurikaan huonompi vaihtoehto opetusikäyttöön kontekstissa, muttei edusta hyviä käytänteitä.

CentOS

Tehdään käyttöjärjestelmään hallinnoija. Komentoriville syötetään yksinkertainen komento:

```
useradd hallinnoija
```

Tässä "useradd" on käyttäjän luontikomento ja "hallinnoija" on annettu argumentti palvelimen käyttöjärjestelmän hallinnoijan nimeksi. Tämän voi luontaisesti halutessaan vaihtaa oman mielen mukaan, mutta asiasta tulee pitää oma-toimisesti myös kirjaa.

Annetaan salasana juuri luodulle tilille:

```
passwd hallinnoija
```

Tilien salasanat tallennetaan kryptattuina tiedostoon /etc/shadow. Jos tilille ei ole annettu salasanaa, tiliä ei voi käyttää, ja tällöin shadow-tiedostossa on tyhjää tilin kohdalla. Salasanallinen tili omistaa pitkän rivin erilaisia merkkejä.

Tekstitiedostot, kuten edellä mainitut passwd ja shadow, voi lukea eri komennoin. Voimme esimerkiksi tarkistaa edellä mainitut tiedostot "cat", "tail" tai "nano" komennoilla. "Tail" palauttaa tekstitiedoston viimeisimmät rivit, mikä tuottaa lyhyemmän syötteen komentotulkkiin, siinä missä "nano" antaa tekstinkäsittelyohjelman käyttökokemuksen. Alla tarkistamme nopeasti, että luomamme tili on käyttövalmiina:

```
tail -n 3 /etc/shadow
```

Annoimme option (kutsutaan myös valitsimeksi) "-n", joka rajaa, kuinka monta riviä tulostetaan komentotulkkiin. Arvo "3" tulostaa 3 viimeisintä riviä.

Käskyjen optiot voi yleisesti tarkistaa "--help" -optiolla. Esimerkiksi äsken käytetyn komennon optiot voi tarkistaa komennolla "tail --help", jolloin komentotulkki tulostaa lisätietoja käskyn käytöstä ruudulle. Help-toiminnallisuus auttaa tukemaan hallinnoijan omavaraisuutta, jolloin kaikkea ei tarvitse tarkistaa muista lähteistä, kuten internetistä.

Lisätään käyttäjä "wheel"-ryhmään, jonka jäsenillä on hallinnolliset sudo-oikeudet. Sen jälkeen id-komennolla voi varmistaa, että käyttäjä löytyy wheel-ryhmästä:

```
usermod -aG wheel hallinnoija
id hallinnoija
```

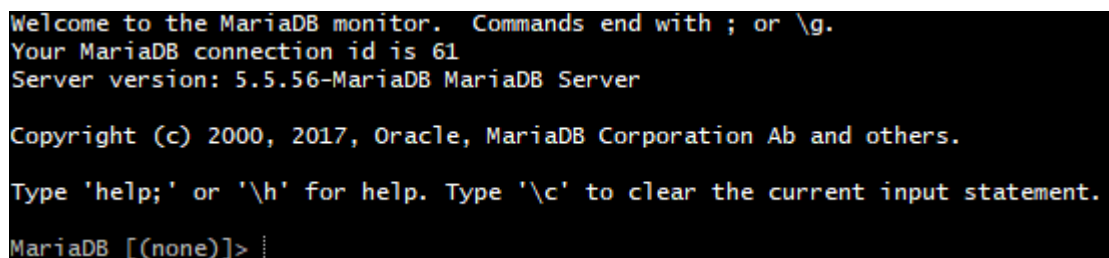
Tietokannan hallinnoija

Tehdään MariaDB:hen hallinnointia varten oma käyttäjätili, jotta hyvien käytäntöjen mukaisesti vältetään käyttämästä juurikäyttäjää ylimääräisesti. Aluksi kirjautumme komentotulkin kautta tietokantaohjelmaan komennolla:

```
mysql -u root -p
```

Kirjautumiskomentoon voi suoraan kirjoittaa salasanan, mutta se lienee varsin huono käytäntö, koska komentorivi itsessään ei ole salattu ja siinä ajettut komennot jäävät salaamattomina talteen. Antamamme komento ilman salasanaa aiheuttaa sen, että ohjelma kysyy käyttäjältä jälkikäteen salasanaa (palvelimen salasana), jonka voi antaa, jotta päästään jatkamaan.

Jos kaikki meni nappiin, pääsimme kirjautumaan tietokannan juurikäyttäjänä MariaDB:hen. Komentotulkin näkymän pitäisi nyt näyttää myös erilaiselle syöterivin kohdalla onnistuneen sisäänkirjautumisen jälkeen. Tilanteen pitäisi pitkälti muistuttaa kuvaa 18.



```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 61
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> :
```

Kuva 18. Komentotulkin komentorivi on muuttunut tietokantajärjestelmään kirjautumisen johdosta.

Teemme hallinnollisen käyttäjätilin ja sen jälkeen annamme sille oikeuksia. Huomiona on, että alla olevat komennot syötetään siis MariaDB:n, eikä käyttöjärjestelmän, komentoina. Tässä kohtaa on syytä varmistaa, että ohjelmassa olisi kuvan 18 tilanne; etenkin, että komentorivi on näyttää erilaiselle (ja siinä lukee "MariaDB").

Alla on yleinen syntaksi, johon valitaan oman mielen mukaisesti nimi ja salasana. Taulukko 2 kokoaa sen parametrit yhteen. Lopun puolipiste kuvastaa komennon loppua:

```
CREATE USER 'username'@'host' IDENTIFIED BY 'password';
```

Taulukko 2. Tietokannan käyttäjätilin luonnin parametreja.

username	webERP-tietokannan hallinnoijan nimi
host	mistä yhteys muodostetaan
password	omavalintainen salasana tilille

Esimerkki käyttäjätilin nimestä voisi olla "hallinnoija", jota käytetään jatkossa. Host määritetään "localhost" -arvolla. Näin ollen koodi olisi (huomaa vaihtaa salasana):

```
CREATE USER 'hallinnoija'@'localhost' IDENTIFIED BY '<salasana>';
GRANT ALL ON weberp.* TO 'hallinnoija'@'localhost' IDENTIFIED BY '<salasana>'
WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

Edeltäneessä koodissa viittaamme tietokantaan nimeltä "weberp". Huomionarvoista on, että muistamme käyttää jatkossa asennusohjelman oletusarvona tarjoamaa tietokannan nimeä "weberp", kuten yllä, jos käytämme tunnollisesti omaa hallinnoijan emmekä juurikäyttäjän tiliä.

4.4.3 webERP:n asentaminen palvelimelle

Seuraavaksi asennetaan tai päivitetään lisäohjelmia käyttöjärjestelmään sekä ladataan netistä webERP:n asennustiedostot palvelimelle. Koodia selitetään auki tässä alaluvussa, mutta siitä muodostettiin pelkistetty versio insinööriyön loppuun liitteeksi, josta sen voi kokonaisuudessaan kopioida ja liittää komentotulkkiin. Kyseisen koodin voi siis maalata ja kopioida tästä asiakirjasta ja siirtyä komentotulkkiin, johon se liitetään hiiren oikealla painikkeella klikaten, jolloin valitaan liitä tai paste, tai näppäimistön näppäinyhdistelmällä, josta Shift+Ins (tai Ctrl+V, tulkista riippuen) tuottanevat halutun lopputuloksen. Enter-painiketta voi joutua painamaan, jos käskyt eivät menneet liitettäessä jo toteutukseen. Enterin ylimääräinen painaminen tuskin aiheuttaa muuta kuin mahdollisen rivinvaihdon komennon suorittamisen jälkeiseen näkymään.

Käyttäjän tulee olla koodirivien kanssa tarkkaavainen, silloin kun ne ulottuvat yli yhden rivin näissä ohjeissa. Lienee toimiva käytäntö kopioida ja liittää niin paljon koodia kuin koodirivejä on kirjoitettu riveittäin putkeen. Havainnointia helpottaa typografia, sillä koodi on indikoitu erilaisella fontilla. Toisin sanoen, kopioidaan kerralla niin paljon koodia, kunnes fontti muuttuu. Käskyjen käsinkirjoitusta ei tekisi mieli suosittaa, sillä silloin saattaa koodiin päästä kirjoitusvirheitä.

Aloitetaan koodin syöttäminen. Ensin suljetaan tietokantajärjestelmä sekä netti-palvelin päivitysten ajaksi:

```
systemctl stop mariadb  
systemctl stop httpd
```

Asennetaan tarvittavia lisäohjelmia ja päivityksiä:

```
yum makecache fast
yum install -y policycoreutils policycoreutils-python selinux-policy selinux-
policy-targeted libselinux-utils yum-priorities yum-utils epel-release
setroubleshoot-server setools setools-console mcstrans
yum -y update
```

Ladataan webERP:n tiedostoja nettipalvelimemme kansioon:

```
cd /var/www/html
git clone https://github.com/webERP-team/webERP.git
```

Hallinnoidaan käyttöoikeuksia. Nettipalvelin tarvitsee oikeuksia webERP:n kansioon:

```
chown -R root:apache /var/www/html/webERP
chmod -R 770 /var/www/html/webERP
find /var/www/html/webERP/. -type f -exec chmod 770 {} \;
```

Edellä annetut oikeudet eivät riitä. Seuraavaksi annamme vielä semanage-alkuisia komentoja, jotka ovat pitkiä ja ulottuvat paikoitellen yli yhden rivin. Tämän takia alla oleva koodi esitetään poikkeuksellisesti rivinvaihtojen kera. Huomattavaa on sanan "companies" esiintyminen rivinvaihtoa edustavalla viivalla, jonka ei pidä siirtyä tekstiä liittäessä tulkkiin. Bashilla ei näin käynyt, mutta tarkkana saa olla:

```
semanage fcontext -a -t httpd_sys_rw_content_t "/var/www/html/webERP(/.*)?"
restorecon -R -F /var/www/html/webERP

semanage fcontext -a -t httpd_sys_rw_content_t "/var/www/html/webERP/compa-
nies(/.*)?"

restorecon -R -F /var/www/html/webERP/companies
```

Seuraavaksi annetaan ohjeita tietokantajärjestelmän päivityksestä. Työssä tehtiin päätös asentaa spesifi versio, sillä se on ainakin ollut yhteensopiva toiminnan kannalta. Huomionarvoista on, että cat-komento ja "EOT" luovat konfigurointitiedoston palvelimelle annetussa muodossaan:

```
cat << EOT >> /etc/yum.repos.d/MariaDB.repo
# MariaDB 10.5 CentOS repository list - created 2020-07-25 15:10 UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name=MariaDB
baseurl=http://yum.mariadb.org/10.5/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
EOT
```

Halutessaan tiedoston luonnin onnistumisen voi varmistaa alla olevalla käskyllä, jonka pitäisi tulostaa edellä mainittua tekstiä:

```
cat /etc/yum.repos.d/MariaDB.repo
```

Annoimme ohjeet asentaa tietty tietokannan versio. Tehdään sama PHP:n kanssa ja asennetaan molemmat maaliin:

```
yum install -y postfix php-mysql
yum upgrade -y mariadb
yum install -y http://rpms.remirepo.net/enterprise/remi-release-7.rpm
yum-config-manager --enable remi-php74
yum install -y php php-cli php-fpm php-mysqlnd php-zip php-devel php-gd php-mcrypt php-mbstring php-curl php-xml php-pear php-bcmath php-json
```

Riittävät päivitykset on tehty. Käynnistetään asennuksen alussa suljetut tietokantajärjestelmä ja nettipalvelin uudelleen:

```
systemctl restart mariadb
systemctl restart httpd
```

webERP:n asennusohjelma

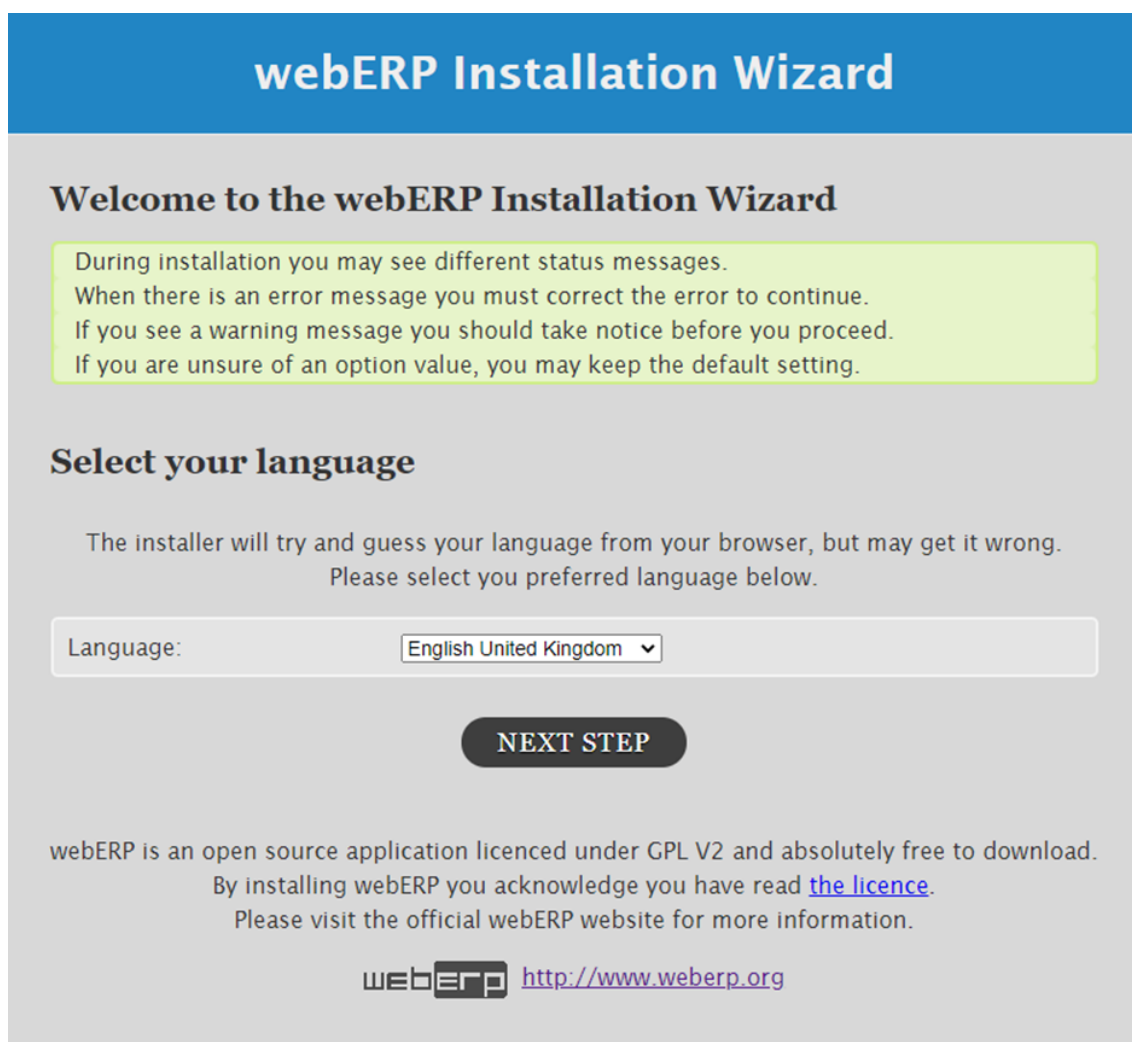
Palvelin on valmisteltu ja webERP:n tiedostot ovat nettipalvelimella valmiina. Nyt voimme viimeistellä asennuksen webERP:n asennusohjelman kautta. Tätä varten menemme omavalintaisella nettiselaimella, kuitenkin edelleen laitteella, joka on Metropolian verkossa, tietohallinnon sähköpostista saatuun IP-osoitteeseen. URL, joka syötetään selaimen, on muotoa: "<IP-osoite>/webERP". On tärkeää huomata, että kirjainkoolla on väliä; webERP:tä ei voi kirjoittaa esimerkiksi kokonaan pienillä kirjaimilla.

Paitsi IP:n, samalla kokonaisen osoitteen saa myös komentotulkista esiin seuraavalla komennolla:

```
echo $(hostname -i)/webERP
```

Yllä olevan syötteen pitäisi tulostaa komentotulkkiin URL, jonka voi kopioida ja liittää nettiselaimen. Ainakin Bashin komentotulkki kopioi tekstin automaattisesti leikepöydälle pelkästään, kun tulkin näkymässä maalaa hiirellä tekstiä.

Kun URL syötetään selaimen, tulisi näkymän näyttää alla olevan kuvan 19 kaltaiselta:

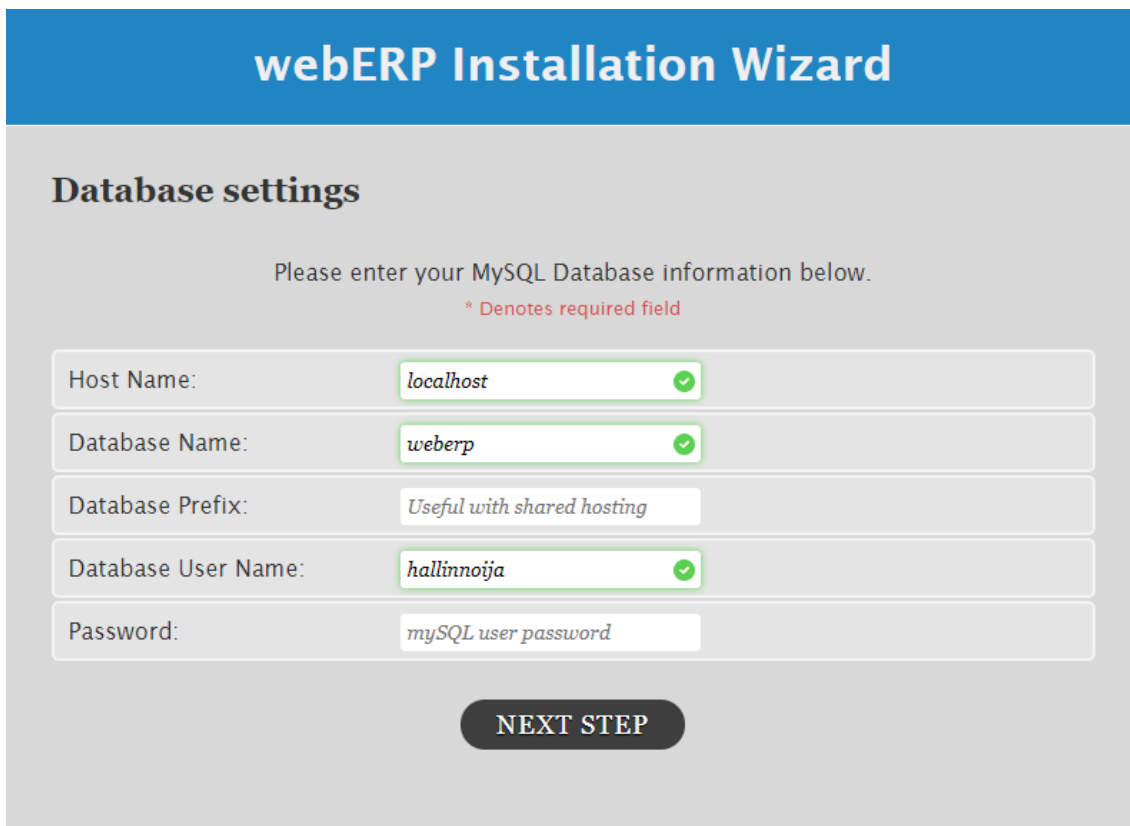


Kuva 19. webERP:n asennusohjelman aloitussivun näkymä.

Käyttäjä voi tässä vaiheessa vaikuttaa vain asennettuun kieleen. Suomen kieltä ei ollut valittavissa insinööriyön aikaan. Tässä kohtaa voidaan painaa selainäkymän painiketta "NEXT STEP".

Jos jotain olisi pielessä, kohdattaisiin virheilmoitus. Virheen kohdatessa asennusta ei voi enää jatkaa, mutta sitä ei tarvitse välttämättä keskeyttää; ohjelmassa on vaatimusten uudelleentarkistus-painike, jota voi painaa, jos virheilmoituksen mukaisen virheen saa korjattua. Yleisin virhe saattaa olla nettipalvelimen käyttäjätilin käyttöoikeuksien puutteet käyttöjärjestelmässä liittyen webERP:n kansioon.

Jos asennusohjelman tarkistus läpäistään, päästään seuraavalle sivulle kuvan 20 mukaisesti.



The screenshot shows the 'webERP Installation Wizard' interface. At the top, there is a blue header with the text 'webERP Installation Wizard'. Below this, the section is titled 'Database settings'. A message reads: 'Please enter your MySQL Database information below.' followed by a red asterisk and the text '* Denotes required field'. There are five input fields, each with a label on the left and a value in the input box, and a green checkmark icon on the right of the input box. The fields are: 'Host Name:' with 'localhost', 'Database Name:' with 'weberp', 'Database Prefix:' with 'Useful with shared hosting', 'Database User Name:' with 'hallinnoija', and 'Password:' with 'mySQL user password'. At the bottom center, there is a dark grey button with the text 'NEXT STEP' in white capital letters.

Kuva 20. Asennus pyytää tietokantajärjestelmään liittyviä tietoja.

Ohjekirjassa mainittiin, että monet kentät olisivat oletusarvallisesti täytetty oikein. Kuvassa 20 määritettiin "Database User Name" oletusarvallisesta "root":sta aiemmin luoduksi tietokantajärjestelmän "hallinnoija":ksi. Jos aiemmin päätettiin ohittaa hallinnointitilien luomiset, käytetään nyt root-tiliä, jonka salasana annetaan tietohallinnon antama salasana.

Painikkeesta "NEXT STEP" pääsemme viimeiselle sivulle asennusta, saapuen kuvan 21 mukaiseen näkymään.

webERP Installation Wizard

Company Settings

* Denotes required field

Company Name: ✔

Chart of Accounts: ▼

Time Zone: ▼

Company logo file: No file chosen jpg file to 10k, not greater than 170px x 80px

Installation option

Install the demo data? WebERPDemo site and data will be installed

Administrator account settings

The default user name is 'admin' and it cannot be changed.
The default password is 'weberp' which you can change below.

webERP Admin Account:

Email address: ✔

webERP Password: ✔

Re-enter Password: ✔

Kuva 21. Asennusohjelman viimeinen vaihe.

Ensimmäiseen kenttään "Company Name" voi määrittää yrityksen nimen, käyttämättä kuitenkaan joitain erikoismerkkejä. Toisen kentän valintaruutuun "Chart of Accounts" ei tarvitse tehdä muutoksia, varsinkaan, jos ei tiedä, mitä tavoittelee.

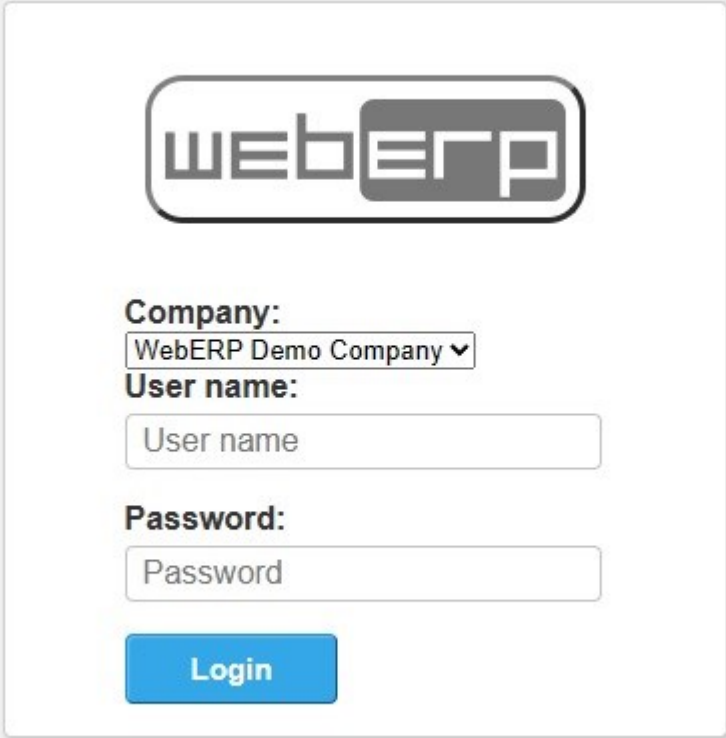
Aikavyöhykkeen ja yrityksen logon voi asettaa halutessaan. Kuvan koko on rajattu varsin pieneksi.

Valintaruutu "Install the demo data?" asentaa valittaessa demodataa webERP:hen. Valinnalla ei ole hirveästi väliä, koska sisäänkirjautuminen tapahtuu yrityskohtaisesti ja demodatan asennus vain tuo yhden vaihtoehdon lisää valittavaksi. Jos demodataa ei tarvitse, voi valinnan poistaa.

Sähköpostiosoitteen antaminen tuskin lienee oleellista tässä kontekstissa. Sen voisi jättää oletusarvolliseksi, ellei ole aiottua suunnitelmaa.

WebERP-järjestelmän tunnusta "webERP Admin Account" ei voi vaihtaa pois oletusarvollisesta "admin":sta, mutta salasanan voi, ja se tuleekin vaihtaa. Luonnollisesti salasana tulisi jatkossa myös itse muistaa.

Kun valintoihin ollaan tyytyväisiä, painetaan painiketta "INSTALL", jolloin asennus päättyy sisäänkirjautumisnäkyymään kuvan 22 mukaisesti.



The image shows a login interface for webERP. At the top is the webERP logo. Below it, there is a 'Company:' label and a dropdown menu showing 'WebERP Demo Company'. Underneath are labels for 'User name:' and 'Password:' with corresponding text input fields. At the bottom is a blue button labeled 'Login'.

Kuva 22. webERP on asennettu ja järjestelemään voi kirjautua sisään.

Järjestelmä on nyt asennettu ja siihen voi kirjautua sisään valitsemalla yrityksen pudotusvalikosta, antamalla käyttäjänimeksi "admin" ja salasanaksi äsken luotu webERP-järjestelmän salasana.

Käyttäjien lisääminen

Samaan toiminnanohjausjärjestelmään luonnollisissakin tilanteissa tulisi päästä useiden eri henkilöiden ja täten useiden eri käyttäjätilien. Vaikkei webERP:n käyttämiseen ei oteta insinööriyössä kantaa, pohditaan lopuksi vielä käyttäjien hallintaa hyvin lyhyesti.

Käyttäjien lisääminen tietokantajärjestelmän kautta on toki eräs vaihtoehto, mutta järjestelmään voi lisätä käyttäjiä kuvan 23 mukaisesti navigoiden päänäkyvästä vasemmalta valiten ensin "Setup" ja sitten "Users Maintenance".

Tätä kautta päästään lomakkeen täyttämistä muistuttavaan näkymään, jossa voi antaa järjestelmään lisättävän käyttäjän tunnukset sekä etenkin käyttäjä-rooli, ”Security Role”.

weberpdemo Demonstration user
Main Menu

Sales	General Setup Options
Receivables	• Company Preferences
Purchases	• System Parameters
Payables	• Users Maintenance
Inventory	• Maintain Security Tokens
Manufacturing	• Access Permissions Maintenance
General Ledger	• Page Security Settings
Asset Manager	• Currencies Maintenance
Petty Cash	• Tax Authorities and Rates Maintenance
Setup	• Tax Group Maintenance
Utilities	• Dispatch Tax Province Maintenance
	• Tax Category Maintenance
	• List Periods Defined
	• Report Builder Tool
	• View Audit Trail
	• Geocode Maintenance
	• Form Designer
	• Web-Store Configuration
	• SMTP Server Details
	• Mailing Group Maintenance

Kuva 23. Käyttäjien hallinta (”Users Maintenance”) löytyy ”Setup”-valikosta.

Palvelimen yhteyden sulkeminen

SSH-yhteys palvelimeen voidaan asennuksen jälkeen sulkea näppäinyhdistelmällä komentotulkissa näppäinyhdistelmällä CTRL+D tai komennolla:

```
logout
```

Yhteys katkennee myös itsekseen tietyn ajan päästä, jos komentotulkki on ollut pitkään käyttämättä. Yhteyttä ei tarvitse ylläpitää komentotulkin kautta, jotta we-

webERP:tä voi käyttää. Laitteen, joka käyttää ohjeiden mukaan asennettua webERP:tä, tulee kuitenkin olla Metropolian verkossa. AnyConnectin VPN-yhteys on helppo ratkaisu yhteyden muodostamiseen verkon ulkopuoliselta koneelta.

4.5 Asennustyökalu

Kun webERP:n asennusohjeiden logiikka oli kehittynyt ensimmäisiin onnistuneisiin versioihin asti, pyrittiin oppimiskokemuksena kehittää työn päätavoitteen, asennusohjeiden, ohella asennustyökalua, joka hoitaisi asennustyön mahdollisimman automaattisesti hallinnoijan puolesta. Näkemys oli minimoida mahdollisten inhimillisten virheiden muodostumista kesken asennuksen sekä nopeuttaa asennusprosessia.

Lopullinen näkemys ohjelman aiotusta (liiallisesta) laajuudesta vahvisti päätöstä olla kehittämättä ohjelmaa loppuun asti. Eräs ongelma oli siinäkin, ettei kaikkea voisi automatisoida, vaan käyttäjänkin syötettä tarvittaisiin aavistuksen verran joka tapauksessa. Tunnollinen kartoitus voisi omata suunnitelman, jonka tekemiseen, esimerkiksi käyttäjätarinoiden tai jopa -haastatteluiden osalta, menisi vääjäämättä aikaa, jota ei suhteessa hyötyyn voinut kunnolla perustella.

Onneksi asentamisen helpottamiseen löytyi hyvä kompromissi. Asennusohjeita parantaessa sekä komentotulkista lisää oppiessa, havaittiin, että tiivistetyt ohjeet voitiin myös kopioida ja liittää useana käskynä putkeen tekstinä komentotulkkiin ilman skriptiäkin. Tämä tehosti manuaalista asennustyötä paljon ja oli viimeistäänkin viimeinen niitti sille, ettei asennustyökalua ollut perusteltua jatkaa.

Kehitystyö tarjosi oppimiskokemuksia, mutta projektin läpimeno alkoi tuntua paljon tärkeämmälle seikalle, kun insinööriyön edistäminen oman arjen lomassa oli turhan satunnaista muutenkin. Eräs hyvä tekniikka oli MVP, josta mainittiin alaluvussa 3.6.2. MVP kannusti rajaamaan nopeasti paisunutta työlistaa. Tajuttiin, että kehityksen alla olevasta työlistasta pitää siirtää valtaosa ohjelmoitavasta toiminnallisuudesta aluksi pois talteen, johon palattaisiin myöhemmillä kehityksiteraatioilla.

Työssä kehitettiin CentOS-alustalla erilaisia skriptejä ohjelman eri osia varten. Rakenteeksi muodostettiin pääohjelma omaan skriptiin, jossa olisi ikään kuin tekstipohjainen, komentotulkkiin ilmestyvä käyttöliittymä käyttäjää varten kuvan 24 mukaisesti.

```
$ ./valikko.sh

***
***  webERP-hallinnointi  ***
***
**                               **
*                               *
*   Valitse toiminto:   *
*   .....   *
*                               *
* [1] - Asenna ympäristö *
* [2] - Testi             *
* [3] - Päivämäärä       *
*                               *
*   Lopetus: ANY KEY     *
*                               *
**                               **
***
***
Valinta: .....
```

Kuva 24. Ohjelman päävalikon kehitysversiona työstettiin omaa skriptinä, josta olisi kutsuja muihin ohjelmatiedostoihin.

Kuvassa 25 on alustavaa hahmottelua ohjelman virtauksesta mock-up -henkeen. Tässä kohtaa toiminnallisuus oli rajattua ja keskityttiin enemmänkin aistiin miltä asennuksen virtaus vaikuttaa.

```
Asennetaan paketteja:
      yum-utils      --> OK
      epel-release  --> OK
      phpmyadmin    --> OK
      polycoreutils-python --> OK

Luodaan käyttöjärjestelmän (CentOS) hallinnoija...
Luodaan tietokannan (MariaDB) hallinnoija...

Luotiin tunnuksiksi:
  CentOS: weberp
  MariaDB: weberp

Asennetaan webERP...
```

Kuva 25. Mock-up asennuksen virtauksesta. Tässä vaiheessa asennusprosessia ei oltu määritetty kokonaan.

Kuvan mock-up -versiosta puuttui esimerkiksi asennettavia lisäohjelmia eikä se vielä kysynyt käyttäjältä hallinnointitilien nimiä tai salasanoja, saati tarjonnut generoimaan satunnaista salasanaa käyttäjän puolesta.

5 Yhteenveto

Retrospektio

Ketterän kehityksen manifesti näkee retrospektion (toiselta nimeltä retrospektiivi) osana ketterää mentaliteettia. Siinä on tarkoitus reflektoida tehtyä työtä ja työtapoja keskittyen tunnistamaan mahdollisia kehityskohteita. Eli tavoitteenaan pohtia kuinka parantaa työskentelytapoja jatkoa ajatellen. Kyseessä voi sanoa olevan jatkuvan kehittämisen ajatusmalli, koska retrospektiota tulisi tehdä tasaisin väliajoin. (Beck 2001: Principles behind the Agile Manifesto.)

Palvelimen tilaamiseen ja ohjelmistojen asennuksiin liittyvät haasteet olivat sopivasti mitoitettuja ja projekti oli sopiva ja mielekäs tältä osin. Teorian kanssa otettiin jälkikäteen ajateltuna turhan laajaa kartoitusta. Työ eteni hajanaisesti, edeten silloin tällöin. Sekä henkilökohtaisesti itseään, että projektin laajuuden rajaamista olisi pitänyt pystyä johtamaan paremmin. Kärsittiin keskittymiseen liittyvistä ongelmista ja energiaa meni hukkaan projektinhallintaan sen elävän tavoitteellisuuden kanssa.

Hyppy tuntemattomaan on vaikeasti ennustettava prosessi. Selkeä suunnittelu vaatisi asiantuntijuutta. Oppimiskokemukset ovatkin tärkeä osa insinööriä. Parempi motivaatio ja yleinen fokus olisivat varmaan olleet oleellisimpia ajureita insinööriä läpimenon vauhdittamisessa.

Ylimääräinen tavoite asennustyökaluun liittyen vei myös aikaa. Sen parissa oli antoisia oppikokemuksia, mutta olisi kenties ollut parempi ensin luoda MVP-versio koko insinööriä, jota olisi sitten voinut lähteä laajentamaan halutessaan. Nyt oli liian monta rautaa tulella, mikä varasi resursseja pois työn etenemisestä. Pidettäkään mielessä siis myös ketterien menetelmien lomassa opittu työn rajoittaminen. Koko projekti oli näin ollen myös oppikokemus henkilökohtaisiin työtapoihin samalla kun sen aikana opittiin tietoteknisiä ratkaisuja.

Kommunikointi tilaajan suuntaan oli omalta osalta turhan laiskaa, toisaalta tähän arvioon vaikuttaa oma subjektiivinen näkemys, joka on monella tapaa kriittinen. Silti voidaan sanoa, että joskus sorruttiin itsesensuuriin kommunikoinnin osalta, edelleen moninaisista syistä. Kenties ei haluttu häiritä työn tilaajaa liian pienillä asioilla, mutta sitten taas jäätettiin miettimään asioita turhankin pitkäksi aikaa itsekseen. Työn ohjaajat, mukaan lukien tilaajan rooli, vastailivat kyllä viesteihin ja tukea oli saatavilla.

Hallinnoinnista

Työn lopputuloksessa on varmasti löydettävissä parantamisen varaa, toivottavasti vain pienessä, ellei minimaalisessa mittakaavassa. Hallinnoijan tulee aina-

kin huomioida eri ohjelmien versioiden tilanne, sillä tulevaisuudessa nämä ohjeet tulevat vanhentumaan tältä osin. Tällöin voisi ensiksi yrittää asentaa kaikista ohjelmistoista uusimmat versiot ja todeta miten palvelin sitten toimii, tässä työssä kun asennettiin tiettyjä versioita yhteensopivuusongelmien johdosta.

Tilanne voi muuttua toisistaan irrallisten ohjelmistojen kehitystyön lisäksi Educloudin katalogin osalta, sillä sen tarjontaa eittämättä tullaan muuttamaan ennen pitkää tulevaisuudessa.

Lähteet

Agarwal, Ravikant & Umphress, David. 2008. Extreme Programming for a Single Person Team. Conference: Proceedings of the 46th Annual Southeast Regional Conference. 28–29.3.2008. Auburn, Alabama, Yhdysvallat.

Artto, Karlos ym. 2008. Projekttiliiketoiminta (2. painos). WSOY, Helsinki.

Beck, Kent ym. 2001. The Agile Manifesto. Verkkoaineisto. <agilemanifesto.org>. The Agile Alliance.

Hammarberg, Marcus & Sunden, Joakim. 2014. E-kirja. Shelter Island, New York, Yhdysvallat: Manning Publications Co.

Hietaniemi, Jari. 2020. Mikä on Kanban? Verkkojulkaisu. <gofore.com/mika-on-kanban>. 18.3.2020. Luettu: 7.9.2021

Hobson, Jonathan & Pelz, Oliver. 2016. CentOS 7 Linux Server Cookbook (2. painos). Birmingham, Iso-Britannia: Packt Publishing Ltd.

Hämäläinen, Henri. 2016. Tehokas Kanban vai kasa lappuja seinällä? Verkkojulkaisu. <contribyte.fi/2016/12/15/tehokas-kanban-vai-kasa-lappuja-seinalla>. 15.12.2016. Luettu 7.9.2021

Hämäläinen, Henri. 2017. Webinaaritallenne. <youtu.be/mQ-BTWUjI_Y>. 2.6.2017.

insightsoftware [sic]. Julkaisuaika tuntematon. Server Hardware Sizing Recommendations. Verkkoaineisto. <support.jetglobal.com/hc/en-us/articles/219401657-Server-Hardware-Sizing-Recommendations>. Muokattu 31.1.2021. Luettu 19.5.2020.

Keski-Ruismäki, Jukka. 2016. SSH-tunnelointi. Verkkoaineisto. <wiki.metropolia.fi/display/tietohallinto/SSH-tunnelointi>. 19.5.2016. Muokattu 12.3.2021 (Lehto, Kalevi). Luettu 4.6.2020.

Keski-Ruismäki, Jukka. 2018. Opetuskäytön educloud virtuaalipalvelimet. Verkkoaineisto. <wiki.metropolia.fi/pages/viewpage.action?pageId=132852058>. 9.8.2018. Muokattu 18.10.2022 (Avikainen, Tomi). Luettu 20.5.2020.

Kettunen, Jari & Simons, Magnus. 2001. Toiminnanohjausjärjestelmän käyttöönotto pk-yrityksessä: Teknologia lähtöisestä ajattelusta kohti tiedon ja osaamisen hallintaa. Espoo: Valtion teknillinen tutkimuskeskus. Saatavilla e-kirjana: <publications.vtt.fi/pdf/julkaisut/2001/J854.pdf>.

Koski, Raimo. 2010. Linux: Tehokas hallinta. Helsinki: Readme.fi.

Kotimaisten kielten keskus. Julkaisuaika tuntematon. Tuotenimet. Verkkoaineisto. <www.kielitoimistonohjepankki.fi/haku/tuotenimet/ohje/180>. Luettu: 22.5.2023.

Microsoft. Julkaisuaika tuntematon. Mikä on ERP? Verkkoaineisto. <dynamics.microsoft.com/fi-fi/erp/what-is-erp/>. Luettu 9.1.2021.

MySQL Calculator. Julkaisuaika tuntematon. Verkkoaineisto. <www.mysqlcalculator.com/>. Luettu: 19.5.2020

Oracle. Julkaisuaika tuntematon. Configuring the Server. Verkkoaineisto. <dev.mysql.com/doc/refman/5.6/en/server-configuration.html>. Luettu 21.6.2020.

Project Management Institute. 2017. Agile Practice Guide. Newtown Square, Pennsylvania, Yhdysvallat: Project Management Institute, INC.

Savonen, Sonia. 2020. Piileekö erp-järjestelmissä uhka? Verkkoaineisto. <www.tivi.fi/uutiset/piileeko-erp-jarjestelmissa-uhka/2ce693bd-1216-4172-8054-c5ce1db62c37 >. 23.4.2020. Luettu 10.1.2021.

Tittel, Ed. 2022. Cloud ERP for dummies - Oracle Special Edition. John Wiley & Sons, Inc. Verkkoaineisto. <www.oracle.com/a/ocom/docs/oracle-cloud-erp-for-dummies.pdf >. Hoboken, New Jersey, Yhdysvallat.

Liitteet

Tiivistetyt asennusohjeet

Alla oleva koodi on pelkistetty kokonaisuus palvelimen CentOS-käyttöjärjestelmään annetuista käskyistä, jotka annetaan palvelimelle yhteyden muodostamisen ja vapaaehtoisten hallinnointitilien luomisen jälkeen (vertaa esimerkiksi taukko 1).

Koodin voi kokonaisuudessaan kopioida ja liittää komentotulkkiin. Sen ajamiseen voi mennä useita minutteja. Koodin pitäisi lopuksi tulostaa dynaamisesti haettu IP-osoite, johon käyttäjä voi selaimella siirtyä asennuksen seuraavaan vaiheeseen.

```
systemctl stop mariadb
systemctl stop httpd
yum makecache fast
yum install -y polycoreutils polycoreutils-python selinux-policy selinux-policy-targeted libselinux-utils yum-priorities yum-utils epel-release
setroubleshoot-server setools setools-console mcstrans
yum -y update

# webERP:n tiedostojen asentaminen koneelle:
cd /var/www/html
git clone https://github.com/webERP-team/webERP.git
chown -R root:apache /var/www/html/webERP
chmod -R 770 /var/www/html/webERP
find /var/www/html/webERP/. -type f -exec chmod 770 {} \;
semanage fcontext -a -t httpd_sys_rw_content_t "/var/www/html/webERP(/.*)?"
restorecon -R -F /var/www/html/webERP
semanage fcontext -a -t httpd_sys_rw_content_t "/var/www/html/webERP/companies(/.*)?"
restorecon -R -F /var/www/html/webERP/companies

# Tehdään tiedosto "/etc/yum.repos.d/MariaDB.repo":
cat << EOT >> /etc/yum.repos.d/MariaDB.repo
# MariaDB 10.5 CentOS repository list - created 2020-07-25 15:10 UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name=MariaDB
baseurl=http://yum.mariadb.org/10.5/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
EOT
yum install -y postfix php-mysql
yum upgrade -y mariadb
yum install -y http://rpms.remirepo.net/enterprise/remi-release-7.rpm
yum-config-manager --enable remi-php74
```

```
yum install -y php php-cli php-fpm php-mysqlnd php-zip php-devel php-gd php-  
mcrypt php-mbstring php-curl php-xml php-pear php-bcmath php-json
```

```
# Apachen ja MariaDB:n uudelleenkäynnistykset päivityksen jäljiltä:  
systemctl restart mariadb  
systemctl restart httpd
```

```
echo -e "\n\nKoodin ajo on päättynyt.\nVoit viimeistellä webERP:n asennuksen  
menemällä selaimella osoitteeseen:\n\n $(hostname -i)/webERP\n"
```