Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

Sähköisen liiketoiminnan järjestelmät

2017

Antti Aronen

RESURSSIENHALLINTA-OHJELMISTON KÄYTTÄJÄKESKEINEN SUUNNITTELU

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Sähköisen liiketoiminnan järjestelmät

2017 | 37

Antti Aronen

RESURSSIENHALLINTA-OHJELMISTON KÄYTTÄJÄKESKEINEN SUUNNITTELU

Projektit sekä niiden resurssien hallinta ovat erittäin tärkeitä yrityksille, etenkin projekteja pääasiallisina tuotteinaan tarjoaville. Markkinoilla on tarjolla lukuisia ohjelmistoja yrityksen projektien ja niiden resurssien hallintaan, mutta ne ovat usein turhan monimutkaisia yrityksen tarpeisiin.

Opinnäytetyön tavoitteena oli tuottaa hyväksyttävä suunnitelma resurssienhallintaohjelmistosta turkulaiselle LST Group Oy:lle. Suunnitelmaan sisältyi tarkennettu vaatimusmäärittely ja käyttötapauksia, joiden pohjalta toteutettiin ohjelmiston alustava prototyyppi jatkokehityksen pohjaksi.

Opinnäytetyö toteutettiin konstruktiivisella otteella ja siinä käytiin läpi käyttäjäkeskeisen suunnittelun eri menetelmiä ja niiden hyödyntämiskeinoja. Opinnäytetyössä kirjoitettiin ohjelmiston suunnitteluprosessin kulusta ja kohdeyrityksen ohjelmistotarpeesta, luotiin ja priorisoitiin kohdeyritykseltä saadut ohjelmistovaatimukset käyttäjätarinamuotoon, luotiin muutama esimerkkikuvaus ohjelmiston toiminnoista käyttötapausten muodossa, sekä esiteltiin vaatimusten pohjalta luotua ohjelmiston alustavaa prototyyppiä, jonka kehitti erillinen opiskelijaryhmä.

Suunnitteluprojekti tuli päätökseensä ja itse prototyyppi ei lopulta täyttänyt kaikkia sille asetettuja vaatimuksia. Vaatimukset, käyttötapaukset ja prototyypin nykytilan kuvaus saatiin toteutettua. Niiden perusteella ohjelmiston voi tulevaisuudessa kehittää.

ASIASANAT:

Käyttäjäkeskeinen suunnittelu, vaatimusmäärittely, resurssienhallinta, ohjelmisto, prototyyppi

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | e-Business Systems

2017 | 37

Antti Aronen

USER-CENTERED DESIGNING OF A RESOURCES MANAGEMENT SOFTWARE

Projects, and the resources management of projects, are very important to companies. Especially to companies that offer projects as their main product. The software market has plenty of options to choose from that specialize on project and resource management, but they are often too complicated for the needs of a company.

The goal of the thesis was to create a plausible plan for creating a resources management software for the target company. The plan included a specified and prioritized requirements analysis as well as some example use-cases, which were the foundation of the software prototype that was created to serve as a platform for possible future development.

The thesis applied constructive methods and addresses the means and standards and implementation of user-centered design. The thesis describes the process of designing the software and provides explanations on the need for the software from the perspective of the target company. It also contains the initial requirements for the software, the created and prioritized final requirements in the form of user stories, some example use-cases of the software, and description of the current state of the prototype for the software that was created.

The designing project came to an end and the prototype did not meet every requirement. The requirements, use-cases and the description of the state of the software prototype were completed. It is possible to create the software based on them.

KEYWORDS:

User-centered design, requirements analysis, resources management, software, prototype

SISÄLTÖ

KUVAT

TAULUKOT

SANASTO

Aktori	Aktori (actor) edustaa UML-kielessä ohjelmiston ulkopuolista tahoa, joka on ohjelmiston kanssa tekemisissä, esimerkiksi käyttäjää tai toista ohjelmistoa. (Sourcemaking 2017)
Excel	Microsoft Excel on taulukkolaskentaohjelma. (Techterms 2017a)
Laravel	Laravel on PHP-kehys web-applikaatioiden luomiseen. (Github 2017)
Nepton	Nepton on suomalainen työajanhallinnan palveluja tarjoava yritys. (Nepton 2017)
Power BI	Power BI on Microsoftin tarjoama työkalupaketti liiketoiminnan tietojen hallintaan. (Power BI 2017)
S-käyrä	Aluksi kiihtyvällä tahdilla kasvava, mutta loppua kohden hidastuva S-kirjaimen muotoa muistuttava, jotain muuttujaa kuvaava käyrä. (Opetushallitus 2017)
The FIRMA	Turun ammattikorkeakoulun pääosin asiakas- ja kehitysprojekteihin keskittyvä oppimisympäristö, joka kokoaa kaikki Kupittaan ICT-Cityssä sijaitsevat projektioppimisympäristöt. (The FIRMA 2017)
Trello	Trello on internetin välityksellä toimiva ilmainen projektinhallinta-alusta. (Trello 2017)
UML	Unified Modeling Language on standardi ohjelmistojen visualisointiin ja mallintamiseen. (Techterms 2017b)
Visma	Yritysohjelmistoja, IT-projekteja sekä -konsultointia tarjoava yritys. (Visma 2017)
WhatsApp	WhatsApp on maksuton viestintäsovellus älypuhelimille ja tietokoneille. (WhatsApp 2017)

1 JOHDANTO

Projektit ja niiden tehokas läpivieminen ovat liiketoiminnalle oleellisen tärkeitä etenkin aloilla, joissa projektit ovat pääasiallinen tarjottava tuote (Forsberg ym. 2004, 3). Resurssien hallinta projekteissa on yleinen haaste, ja usein syynä on projektin suunniteltu budjetti. Joko ei ole riittävästi resursseja tai oikeanlaista osaamista projektin läpiviemiseen vapaana olevista työntekijöistä, projektin alustava tavoite tai ympäristö muuttuvat esimerkiksi talouden tai teknologian mukaan, tai projektin edetessä suunnitelmassa huomataan puutteita tai muuta korjattavaa. (Kettunen 2009, 162–163)

Projektien ja niiden resurssien hallintaan on markkinoilla tarjolla ohjelmistoja, mutta ne ovat usein kaikessa monipuolisuudessaan myös monimutkaisia, eivätkä välttämättä sovi ohjelmiston käyttöönottoa harkitsevan yrityksen tarpeisiin (Projekti-instituutti 2017). Tämän opinnäytetyön ote on konstruktiivinen. Opinnäytetyön tarkoituksena on suorittaa alustava konseptisuunnitelma resurssienhallinta-ohjelmistosta, joka soveltuu turkulaisen sähköalan konserni LST Groupin projektien ja niiden resurssien hallinnan tarpeisiin yrityksen nykyistä ratkaisua tehokkaammin. Konseptisuunnitelma perustuu ennalta saadun tarvekartoitusdokumentin sisällön tarkentamiseen vaatimuksiksi ja niiden priorisoimiseen. Konseptisuunnitelman perusteella ohjelmiston alustavan prototyypin kehittämisen suorittaa erillinen opiskelijaryhmä, josta opinnäytetyössä myös kirjoitetaan.

LST Group Oy:llä on useita projekteja käynnissä samanaikaisesti, ja yritys tarvitsee työkalun helpottamaan niiden resurssien hallintaa. Heidän nykyisillä tähän tarkoitukseen olevilla työkaluilla ei päästä konsernia tyydyttävään lopputulokseen, sillä työntekijöitä voidaan varata projekteihin, vaikkei käynnissä olevan projektin tarve ole vielä täyttynyt, tai projektipäälliköiden sitoutuminen budjettien päivittämiseen on ollut vaihtelevaa.

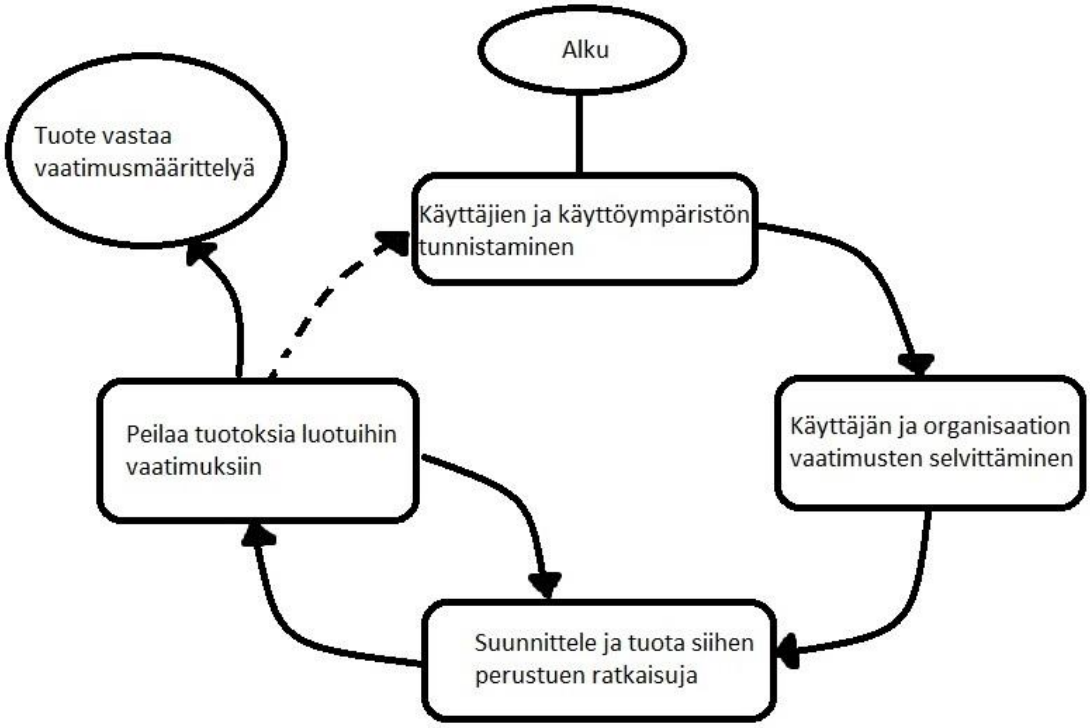
Opinnäytetyön teoriaosuus keskittyy käyttäjäkeskeisen suunnittelun menetelmiin yleisesti, mutta testausta käsittelevä osuus on jätetty lyhyeksi sekä viimeinen vaihe eli ylläpito on rajattu tästä työstä kokonaan ulos. Käytännön osuudessa kerrotaan LST Groupille luodusta ohjelmistosuunnittelusta ja sen perusteella kehitetystä prototyypistä sekä koko prosessin eri vaiheiden kulusta. Lopetusluvussa katsotaan aikaansaatuja tuloksia ja pohditaan suunnitteluprosessin ja prototyypin kehityksen onnistumista sekä jatkokehityksen mahdollisuutta.

2 OHJELMISTON KÄYTTÄJÄKESKEINEN SUUNNITTELU

Käyttäjäkeskeisellä suunnittelulla pyritään saamaan ohjelmistosta sekä käytettävyydeltään että käyttökokemukseltaan mahdollisimman hyvä. Käytettävyydellä tarkoitetaan lopputuotteen käyttölaatua ja käyttökokemuksella käyttäjän kokemuksen laatua lopputuotteen parissa. Molemmat vaikuttavat toisiinsa: puutteet käytettävyydessä heijastuvat käyttökokemukseen ja toisin päin. Hyvä käytettävyys ja käyttökokemus vaikuttavat siis myös positiivisesti liiketoimintaan, sillä asiakas palaa mieluusti käytettävyydeltään hyvän tuotteen pariin, kun taas huono käyttökokemus saa asiakkaan helposti etsimään vaihtoehtoista ratkaisua. (Nichols & Chestnut 2013, 8–12; Sinkkonen ym. 2009, 18–19)

Ohjelmiston käyttäjäkeskeinen suunnittelu pitää käyttäjän keskiössä koko suunnitteluprosessin ajan, jotta lopputuote on käyttäjän kannalta oikeasti tarpeellinen eikä vain suunnittelijan näkemyksen mukainen tarpeellisesta. Tähän lopputulokseen päästään käymällä aktiivisesti keskustelua käyttäjien ja muiden suunnittelijoiden kanssa. (Huotari ym. 2003, 9)

Ohjelmiston käyttäjäkeskeinen suunnittelu on monivaiheinen ja suunniteltavan ohjelmiston kokoluokasta riippuen pitkä prosessi (Kuva 1). Jokainen projekti on erilainen eikä kaikkia käyttäjäkeskeisen suunnittelun vaiheita välttämättä tarvitse hyödyntää samalla tavalla kuin toisessa projektissa (Huotari ym. 2003, 18). Käyttäjäkeskeisen suunnittelun pääasiallinen idea on kuitenkin läsnä jokaisessa projektissa: ensin selvitetään ohjelmiston mahdollisten käyttäjien luonne, käyttötapa ja käyttöympäristö. Sen jälkeen kehitetään ohjelmistoa saadun tiedon perusteella ja testataan, onko saavutettu tulos toimiva. (Sinkkonen ym. 2009, 33) Seuraavaksi käydään läpi ohjelmiston käyttäjäkeskeisen suunnittelun vaiheita, joita hyödyntämällä ohjelmistoa kehitetään eteenpäin asteittain, kunnes sille asetetut vaatimukset täyttyvät.



Kuva 1. Käyttäjäkeskeisen tuotekehityksen idea (Sinkkonen ym. 2009, 34).

2.1 Vaatimusten määrittely

Ohjelmistoprojektin aluksi on syytä selvittää tarkat syyt sille, miksi ohjelmistoa ylipäätään tarvitaan. Onnistuessaan ohjelmisto täyttää sekä liiketoiminnalliset että käyttäjätarpeet (Sinkkonen ym. 2009, 51). Nämä saadaan selville suorittamalla vaatimusten määrittely, joka muodostuu vaatimusten määrittelyyn valmistautumisesta, vaatimusten tuottamisesta ja vaatimusten määrittelyn hyväksymisestä. Valmisteluvaiheessa tehdään projektisuunnitelma määrittelyn läpiviemiseksi. Suunnitelmassa kuvataan vaatimusten määrittelyn toteutustapa, aikataulu ja toteuttajat. (JHS 2012)

Vaatimusten tuottamiseksi käytetään useita keinoja, missä analysoidaan projektin liiketoiminnallisia perusteita, tutkitaan ohjelmiston käyttäjäkuntaa – niin nykyisiä kuin tuleviakin – ja määritellään ohjelmiston toiminnalliset ja ei-toiminnalliset vaatimukset. Myös ohjelmiston tietovaatimuksia kerätään esimerkiksi käsitemallin avulla (JHS 2012). Seuraavaksi käsitellään lyhyesti eri vaatimustyyppejä ja niiden selvittämisen keinoja.

2.1.1 Liiketoiminnalliset vaatimukset

Tärkein lähtökohta ohjelmistoprojektille asiakkaan kannalta on sen liiketoiminnallinen perusta, joten projektia tarkastellaan asiakkaan puolesta usein liiketoiminnallisen hyödyn näkökulmasta (Haikala & Mikkonen 2011, 31). Liiketoiminnalliset vaatimukset saadaan yleensä asiakkaalta. Kaikki liiketoiminnalliset vaatimukset pyrkivät pohjimmiltaan rahallisen hyödyn saavuttamiseen asiakkaalle, joka saavutetaan joko tuottoja lisäämällä tai kustannuksia vähentämällä. (Sinkkonen ym. 2009, 51)

Kilpailijavertailu auttaa löytämään ideoita omaan ohjelmistoon kehittämisprosessin alussa. Tarkoituksena ei ole suoraan kopioida kilpailevien ohjelmistojen ominaisuuksia, vaan kartoittaa mahdollisia löytyviä kilpailuvaltteja ja puutteita. Näiden löytöjen avulla on helpompaa valita ja priorisoida oman ohjelmiston kehitettäviä osia. Kilpailijavertailun voi toteuttaa itse tai tilata asiaan perehtyneeltä yritykseltä vertailututkimuksen. (Sinkkonen ym. 2009, 56–57)

Tarveanalyysihaastattelulla voi selvittää liiketoiminnallisia tarpeita, asiakastarpeita ja pohjustaa tulevaa käyttäjätutkimusta. Haastattelun kohteina voivat olla organisaation päättäjät, myynti- ja markkinointihenkilökunta, asiakkaat ja käyttäjät. (Sinkkonen ym. 2009, 59)

2.1.2 Käyttäjätutkimus

Liiketoiminnallisten vaatimusten selvittämisen jälkeen siirretään huomio ohjelmiston käyttäjiin. Käyttäjätutkimuksessa hankitaan tietoa käyttäjien tarpeista koskien heidän tavoitteita, tehtäviä, rajoituksia, motiiveja sekä olosuhteita. Käyttäjätutkimuksen toteutusmuoto voi vaihdella riippuen ohjelmiston koosta, tietotarpeista, käytössä olevasta budjetista ja aikataulusta. Käyttäjätutkimus voi siis olla joko muutaman tunnin haastattelu tai useita viikkoja kestävä havainnointitutkimus. (Sinkkonen ym. 2009, 65)

Käyttäjiin luetaan ohjelmistoa suoraan käyttävät henkilöt, ylläpitäjät, mahdolliset ohjelmiston tukihenkilöt sekä tilastoja ja käyttöä seuraavat henkilöt. Käyttäjät ryhmitellään samankaltaisten tavoitteiden ja tarpeiden, sekä erottavien tekijöiden mukaan luomalla käyttäjäprofiileja. Näitä ominaisuuksia ovat osaaminen, toimintatapa, tarpeet, roolit, kokemus, toimintaolosuhteet sekä käyttäjän ikä. Yksi käyttäjäryhmä valitaan ensisijaiseksi, ja suunnittelu toteutetaan sille muita ennen. (Sinkkonen ym. 2009, 66–67)

Käyttäjäryhmien löytämiseksi on useita tapoja. Yksi tapa on miettiä, kenelle ohjelmistoa suunnitellaan: ketkä ovat nykyisiä käyttäjiä ja keitä halutaan käyttäjiksi tulevaisuudessa. Nämä tiedot saadaan esimerkiksi haastattelemalla liiketoimintaan sidoksissa olevia henkilöitä, kuten markkinoinnin, myynnin ja muun liiketoiminnallisen kehityksen työntekijöitä. Toinen tapa on miettiä, ketkä saattaisivat tarvita suunnitteilla olevaa ohjelmistoa. On myös hyvä pohtia niitä käyttäjiä, jotka eivät luonnostaan kuulu kohdekäyttäjiin ja kenet jätetään ulkopuolelle. (Sinkkonen ym. 2009, 68–69)

Käyttäjätutkimuksen alussa on hyvä laatia tutkimussuunnitelma, josta selviää tutkimuksen tavoite, aikataulu ja budjetti. Seuraavaksi valitaan tutkimuksen kohteena olevat käyttäjät mahdollisimman monipuolisesti eri käyttäjäryhmistä. (Sinkkonen ym. 2009, 79–81)

Käyttäjätutkimuksen toteutuksessa voidaan käyttää useampia menetelmiä. Näitä ovat haastattelut, kyselyt, havainnointi ja testaus, tarinat, päiväkirjat, roolileikit ja simulaatiot. Menetelmät valitaan sen mukaan, mitä tietoa halutaan saada selville. (Sinkkonen ym. 2009, 70–71)

Samankaltaisuusanalyysiä käytetään käyttäjätutkimuksen aikana saatujen tietojen järjestämiseen hallittaviksi kokonaisuuksiksi. Analyysin aikana samoihin asioihin liittyvä käyttäjäaineisto ryhmitellään keskenään esimerkiksi seinätaululle kokonaisuuksien hahmottamisen helpottamiseksi. (Sinkkonen ym. 2009, 118) Samankaltaisuuksia etsiessä voidaan kiinnittää huomiota esimerkiksi käyttäjien tarpeisiin, käyttäytymismalleihin, haluihin, yleisimpiin ongelmiin, mitkä käyttäjät tuottavat eniten kassavirtaa ja keiden palveleminen on kaikkein kalleinta (Nichols & Chestnut 2013, 42–43).

Persoonat luodaan tiivistämällä tunnistettuja käyttäjäryhmiä. Niitä voidaan hyödyntää toimintatapojen ja suunnitteluratkaisujen määrittelyyn ja vertailuun, roolileikkeihin sekä hahmottamaan käyttökokemuksia. (Sinkkonen ym. 2009, 124–125) Persoonista kehitetään mahdollisen aidon tuntuisia, jotta kohdekäyttäjän maailmaan on helpompi eläytyä. Jotta luodut persoonat eivät paisu liian isoiksi, on syytä lisätä niihin vain kohdekäyttäjän oleelliset ja muista käyttäjistä erottelevat tiedot. (Nichols & Chestnut 2013, 50–52)

2.1.3 Ohjelmiston toiminnalliset ja ei-toiminnalliset vaatimukset

Kun liiketoiminnalliset ja käyttäjätarpeet on selvitetty, siirretään huomio ohjelmiston toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Toiminnalliset vaatimukset kuvaavat ohjelmistolla tehtäviin toimintoihin liittyviä vaatimuksia ja ei-toiminnalliset vaatimukset

kuvaavat ohjelmistossa huomioon otettavia tekijöitä, kuten ohjeistusta, tietoformaatteja ja ylläpidettävyyttä. Jokainen ohjelmiston haluttu toiminto tulee kuvata vaatimuksena. Ei-toiminnallisiin vaatimuksiin lukeutuu ohjelmiston tietovaatimukset, toimintaympäristövaatimukset, käytettävyysvaatimukset, saavutettavuusvaatimukset ja turvallisuusvaatimukset. (JHS 2012; Sinkkonen ym. 2009, 49)

Toimintatarinat luodaan kuvaamaan ohjelmiston nykyisiä toimintatapoja käyttäjätutkimuksesta saatujen tietojen perusteella. Tästä on hyötyä toimintatapoihin suunniteltujen parannusten hahmottamiseksi, mikäli toimintatapojen uudet versiot ovat samantyylisiä kuin vanhat. Toimintatarinat voidaan luoda käyttäjätutkimuksesta tai käyttäjiltä suoraan saatujen tietojen perusteella. Toimintatarinoiden sisältö riippuu täysin siitä, minkälaiseen ohjelmiston toimintaan ne perustuvat. Niiden tulisi kuitenkin kuvastaa yhden käyttäjän tai persoonan toimintatapaa konkreettisella tasolla. (Sinkkonen ym. 2009, 135–136)

Käyttötarinat luodaan toimintatarinoiden ja kaiken aiemmin kerätyn materiaalin perusteella ohjelmiston toiminnallisuuden kuvaamiseksi. Käyttötarinat liittyvät aiemmin luotuihin persooniin ja kuvaa toimintatarinassa kerrotun nykytilanteen tavoitetilaa. (Sinkkonen ym. 2009, 171–175)

Käyttäjätarina on lyhyt kuvaus yhden käyttäjän tai käyttäjätyypin toiminnasta ohjelmiston jonkin osan kanssa. Käyttäjätarinaan tulee kirjata selkeästi toiminnon tekijän identiteetti, sekä mitä ja miksi tehdään. Käyttäjätarinat ovat selkeä tapa kuvata ohjelmiston oleellisia käyttäjävaatimuksia. Käyttäjätarinoita voi ohjelmiston laajuudesta riippuen syntyä runsaasti, joten niiden priorisointi on järkevää, jotta pystytään kohdistamaan huomio oleellisesti tärkeisiin vaatimuksiin. (Tolvanen 2013)

Käyttötapaukset luodaan kuvaamaan käyttäjien vuorovaikutusta ohjelmiston toimintojen kanssa yleisellä tasolla, tekstimuodossa. Yhteen käyttötapaukseen voi sisältyä useampia käyttäjiä ja käyttötarinoita. (Sinkkonen ym. 2009, 181) Käyttötapaukset kirjoitetaan selkeällä ja helposti ymmärrettävällä kielellä, joka on suuri etu asiakkaan kanssa kommunikoidessa. Käyttötapausten perusteella on hyvä miettiä ohjelmiston toiminnallisia vaatimuksia. On kuitenkin syytä varoa toimintojen liian yksityiskohtaista kuvaamista, jotta kuvaukset eivät määrää tulevaa toteutustapaa liiaksi jo määrittelyvaiheessa. (Haikala & Mikkonen 2011, 80)

Käyttötapausten kuvaamiseksi voidaan tehdä sekä käyttötapauskaavio että käyttötapauskuvauksia. Käyttötapauskaavio luodaan käyttäen UML-kieltä (Unified Modeling Language), jossa kuvataan kaavion keskelle käyttötapaukset, reunoille ohjelmiston

käyttäjät eli aktorit ja lopuksi yhdistetään aktorit heille tarkoitettuihin käyttötapauksiin kertoen heidän osallisuudestaan kyseiseen toimintoon. (Haikala & Mikkonen 2011, 77–78) Kaavion avulla pyritään hahmottamaan kaikki mahdolliset käyttäjävaatimukset ja toimintatavat mitä suunniteltavalla ohjelmistolla on olemassa (Huotari ym. 2003, 59).

Käyttötapauskuvaus on puolestaan sanallinen kuvaus käyttötapauksen tapahtumaketjusta. Kuvaukseen merkitään yleensä käyttötapauksen nimi, versio, käyttäjät, alkuehdot, sanallinen kuvaus, poikkeustilanteet, lopputulos ja mahdolliset muut vaatimukset. (Haikala & Mikkonen 2011, 79–80)

2.1.4 Vaatimusten priorisointi ja hyväksyminen

Määrittelyvaiheessa luodaan lukuisia ohjelmiston mahdollisia ominaisuuksia. Niistä kaikki eivät kuitenkaan ole toteutustarpeeltaan samanarvoisia. Osa vaatimuksista on helpompi toteuttaa ja ylläpitää kuin toiset, ja osan toteuttaminen ja ylläpito tulevat kalliimmaksi kuin toisten. Kannattaa siis priorisoida sellaiset ominaisuudet korkeammalle, jotka ovat sekä helppoja että halpoja toteuttaa. On myös syytä miettiä, mitkä ominaisuudet ovat liiketoiminnan ja yleisen ohjelmiston toiminnallisuuden kannalta välttämättömiä, ja mitkä ominaisuudet tuottavat lähinnä lisäarvoa, jotta suunnitteluprosessi pysyy aikataulussa ja tuottaa haluttua tulosta. Vaatimusten prioriteetit voivat vaihtua projektin edetessä. (JHS 2012; Nichols & Chestnut 2013, 107–108)

Vaatimuksia kerätään koko määrittelyprosessin ajan ja niiden perusteella luodaan vaatimusdokumentti. Vaatimuksia voi kerätä myös ulkoiseen järjestelmään tai esimerkiksi Excel-taulukkoon. (Haikala & Mikkonen 2011, 66) Kerätyt vaatimukset käydään läpi järjestämällä katselmointitilaisuus projektin sidosryhmien kesken, jossa tarkastellaan vaatimusten ymmärrettävyyttä, oikeellisuutta ja tarkkuutta. Katselmoinnin tulosten pohjalta mahdolliset korjatut vaatimukset hyväksytetään ohjelmiston omistajalla tai hänen edustajallaan, joka hyväksyy vaatimukset tai palauttaa ne uudelleen käsiteltäväksi. (JHS 2012)

2.2 Suunnittelu

Kun ohjelmiston vaatimukset on määritelty, priorisoitu ja hyväksytty, alkaa ohjelmiston varsinainen suunnitteluvaihe. Ohjelmiston rakenne suunnitellaan tukemaan luotujen

vaatimusten toteuttamiseen vaadittavia ratkaisuja ja tehtäväprosesseja, joiden perusteella luodaan rautalankamalleja visualisoimaan suunniteltuja ratkaisuja. (Nichols & Chestnut 2013, 155; Haikala & Mikkonen 2011, 177)

2.2.1 Informaatioarkkitehtuuri

Informaatioarkkitehtuurin laadukas suunnittelu on erittäin tärkeää, sillä se vaikuttaa ohjelmiston sisällön rakenteisiin ja tätä kautta myös ohjelmiston käyttökokemuksen laatuun tuomalla selkeyttä kokonaisuuden hahmottamiseen. Informaatioarkkitehtuurin suunnittelussa keskitytään tiedon rakenteisiin, organisointiin ja luokitteluun. Ohjelmiston kokonaisrakenteen suunnittelu helpottaa yksityiskohtaisemman sisällön suunnittelemista myöhemmin. (Nichols & Chestnut 2013, 155; Sinkkonen ym. 2009, 183–184)

2.2.2 Rautalankamallit

Rautalankamalleilla luodaan ohjelmiston rakenteen toiminnallinen suunnittelu. Toiminnot selitetään ja perustellaan. Rautalankamalli on usein yksinkertainen kuvaus, joka voidaan tarvittaessa, vaikka piirtää paperille, tai toteuttaa rautalankamallin luomiseen tarkoitetulla ohjelmalla. Useimmat näistä ohjelmista eivät vaadi lainkaan ohjelmointia, joten ne soveltuvat kenen tahansa käytettäväksi. Rautalankamallit ovat nopeasti ja halvalla toteutettavissa, joten ne soveltuvat mainiosti usean erilaisen suunnitteluratkaisun havainnollistamiseen, testaamiseen ja jatkosuunnitteluun. (Nielsen 1993, 93; Sinkkonen ym. 2009, 203) Jotta keskustelu rautalankamallien parissa keskittyy olennaiseen, eli ohjelmiston toiminnallisuuden esittelemiseen, kannattaa rautalankamallien ulkoasu jättää yksinkertaiseksi. (Haikala & Mikkonen 2011, 39)

Jos käytetään paperille luotuja rautalankamalleja kuvaamaan laajan ohjelmiston toiminnallisuutta, on esittelijän syytä perehtyä ohjelmiston tarkoitettuun toimintatapaan huolellisesti, sillä paperirautalankamallien kasautuessa voi tarpeellisen seuraavan ohjelmistonäkymän löytyminen suuresta pinosta hankaloitua. Paperirautalankamallien käyttämisen yksi eduista on niiden ympäristövaatimukset, sillä ne eivät vaadi esittelyn avuksi tietokonetta tai seinälle heijastettua kuvaa. Niitä voidaan myös, niin haluttaessa, luoda asiakkaan kanssa samassa tilassa reaaliaikaisesti. Tämän toimintamallin etuna ovat vielä nopeammat syklit kehityksessä, sillä paperille tehtävät muutokset ovat todella nopeasti tehtävissä. (Nielsen 1993, 97–99)

2.2.3 Visuaalinen suunnittelu

Visuaalisella suunnittelulla luodaan ohjelmiston yleisilme, joka viestii ohjelmiston brändistä ja persoonallisuudesta. Hyvä visuaalinen suunnittelu ohjaa käyttäjän huomion ohjelmiston olennaisiin kohteisiin ilman, että niiden löytämiseen tarvitsee nähdä liikaa vaivaa. Visuaalinen suunnittelu voidaan aloittaa kunnolla vasta, kun rautalankamallit on jo luotu. Rautalankamalleja tehdessä on hyvä kuitenkin olla jo jonkinlainen käsitys siitä, miltä lopullinen visuaalinen ilme tulee näyttämään. (Sinkkonen ym. 2009, 242)

2.3 Toteutus

Suunnitteluvaiheen jälkeen siirrytään ohjelmiston varsinaiseen toteuttamiseen. Rautalankamalleista siirrytään ohjelmoituihin prototyyppeihin, joiden perusteella testataan ja arvioidaan ohjelmiston käytettävyyttä iteratiivisesti, kunnes lopputulos on halutun kaltainen (Hyysalo 2006, 171). Toteutusvaiheen valmistelussa on hyvä käyttää apuna aiemmin luotuja dokumentteja, kuten vaatimusmäärittelyä, jotta pystytään todentamaan tuotettujen ratkaisujen onnistumista (JHS 2016).

2.3.1 Ohjelmiston prototyypin luominen

Ohjelmoitava prototyyppi luodaan, jotta saadaan testattua kehitteillä olevaa ohjelmistoa syvemmin kuin mihin rautalankamallit kykenevät. Prototyypin avulla tulevat käyttäjät saavat konkreettista kuvaa siitä miten suunniteltu ohjelmisto toimii, ja tuo siten selkeyttä käyttäjien visioon kunkin ominaisuuden varsinaisesta toimintatavasta ja tarpeellisuudesta. (Hyysalo 2006, 171)

Ohjelmiston prototyyppi voi olla varsinaisen ohjelmiston esiaste, jonka päälle lopullinen tuote rakennetaan eteenpäin kehittäen, tai se voi olla pois heitettävä ja erillinen kokonaisuus. Prototyypin päälle rakentaessa voi syntyä ongelmia joidenkin ominaisuuksien kanssa, mikäli ne eivät ole optimaalisesti toteutettuja. Projektin edetessä saattaa tulla kiusaus jättää vanha ja puutteellinen, mutta toimiva ratkaisu sellaisenaan ohjelmiston osaksi ajan säästämiseksi tai sitä ei huomata lainkaan testausvaiheessa. (Haikala & Mikkonen 2011, 38–39)

Prototyyppiin tehtävät ratkaisut ja muutokset voidaan kirjata, jotta pysytään kartalla tehdyistä ja hylätyistä ratkaisuista. On myös järkevää kirjoittaa perustelut toimenpiteille, jotta vältytään samojen ongelmakohtien sisällyttämiseltä ohjelmistoon. (Hyysalo 2006, 178)

2.3.2 Käytettävyyden testaus ja arviointi

Käytettävyyttä arvioimalla ja testaamalla selvitetään ohjelmiston julkaisukelpoisuus, ja ovatko määritetyt käytettävyyden vaatimukset täytetty. Käytettävyystesteissä on mukana käyttäjiä, joten se tuottaa luotettavampaa tietoa. Arvioinnit kuitenkin tukevat testejä ja ovat halvempia toteuttaa. (Sinkkonen ym. 2009, 285)

Heuristinen arvio

Heuristinen arvio toteutetaan käymällä läpi ohjelmiston käyttöliittymää virheitä etsien ja korjausehdotuksia antaen (Sinkkonen ym. 2009, 287). Arvio tuo selkeyttä siihen, mitä käyttöliittymän suunnittelija olettaa ohjelmiston käyttäjän näkevän, ja mitä käyttäjä todellisuudessa näkee. Tarkasteltavat asiat pohjautuvat yleensä etukäteen luotuun listaan, joka perustuu joko ennalta julkaistuihin tai itse luotuihin sääntöihin. (Nielsen 1993, 155)

Heuristisen arvion toteuttamiseen osallistuu yleensä useampi arvioija kerrallaan, sillä yksittäiseltä arvioijalta jää todennäköisesti oleellisia virheitä huomaamatta. Eri arvioijat myös usein löytävät eri virheitä. Jokainen arvioon osallistuva tarkastelee ohjelmistoa itsenäisesti ja kommunikoi toisten arvioijien kanssa vasta arviointisession lopuksi, jotta arviot eivät muutu toisten vaikutuksesta. Arviointikierroksia toteutetaan yleensä vähintään kaksi. Arvion tulokset kirjataan raporttiin jokaisen arvioijan toimesta tai vaihtoehtoisesti arviointisessioissa on mukana tarkkailija, joka kirjaa tulokset ylös arvioijien suullisen palautteen perusteella. Kun arviointi on tehty, kokoonnutaan arvioijien ja tarkkailijan kesken yhteen, tarkastellaan löydettyjä virheitä ja pohditaan niihin ratkaisuja. Saadut tulokset kirjataan, jonka jälkeen niitä ryhdytään korjaamaan. (Nielsen 1993, 156–157; Sinkkonen ym. 2009, 288)

Käytettävyystesti

Käytettävyystestissä tarkastellaan ohjelmiston käyttäjän toimintatapaa ja reaktioita mahdollisimman aidon tuntuisessa käyttötilanteessa, jotta saadaan tietoa ohjelmiston käytettävyyden nykytilasta ja löydetään ne käytettävyyden virheet, jotka eivät tulleet esiin heuristisen arvion tuloksena. Testaajat kannattaa valita monipuolisesti ohjelmiston kohdekäyttäjistä (Huotari ym. 2003, 76). Käytettävyystesti toteutetaan ennalta luodun testitarinan perusteella ja sen tulokset tallennetaan ja analysoidaan jatkokehitystä varten. (Sinkkonen ym. 2009, 299)

Käytettävyystestin aluksi tulee tehdä käyttäjille selväksi, että tilanteessa testataan tuotteen käytettävyyttä eikä heidän osaamistaan. Keskustelu testin aikana on sallittua, mutta ohjaaja ei auta ohjelmiston käyttämisessä, jotta testitulos on luotettava ja mahdollisimman hyödyllinen. Testiin osallistuvien käyttäjien taustoista on hyvä selvittää ainakin heidän mahdollinen aikaisempi osaaminen testattavan kaltaisten ohjelmistojen käyttämisessä, ikä ja ammatti. Testin lopuksi pidetään loppuhaastattelu, missä käydään läpi testitilanteessa esiin nousseita asioita. Haastattelun aikana selvinneet asiat kirjataan muistiin jatkokehitystä varten. (Sinkkonen ym. 2009, 306–307)

2.4 Testaus

Toteutusvaiheen jälkeen ohjelmisto on lähellä julkaisukelpoisuutta, mihin pyritään testausvaiheessa. Nyt testattavana on koko järjestelmä. Julkaisukelpoisuuden selvittämiseksi testausvaiheen tuloksia verrataan aiemmin luotuun dokumentaatioon, kuten vaatimusmäärittelyyn ja mahdolliseen käyttöohjeeseen. Testaajiksi kannattaa valita ohjelmiston kehitystyön ulkopuolisia henkilöitä, sillä ohjelmiston kehittäjät saattavat ajatusmalliltaan pyrkiä todistamaan sen toimivuutta virheiden etsimisen sijaan. Testattavia asioita ovat ainakin ohjelmiston selviäminen kuormituksesta, toipuminen virhetilanteista, asentamisen mutkattomuus ja käytännön testaus ohjelmiston tarkoitetussa toimintaympäristössä. (Haikala & Mikkonen 2011, 208–209)

Testitapauksia valittaessa on kaksi peruslähestymistapaa: lasilaatikkotestaus ja mustalaatikkotestaus. Lasilaatikkotestauksessa hyödynnetään tietoa ohjelmiston toteutuksesta, eli testaajilla on ohjelmakoodi käytettävissään. Mustalaatikkotestauksessa ei

kiinnitetä huomiota ohjelmiston rakenteeseen tai koodiin, vaan keskitytään sen määritelmiin. (Haikala & Mikkonen 2011, 209; Joensuun yliopisto 2007)

Testitapausten valinnan yhteydessä on syytä luoda testaussuunnitelma, johon kirjataan myös testauksen hyväksymiskriteerit ja lopettamiskriteerit, jotta tiedetään testauksen lopettamiselle sopiva ajankohta. Näitä kriteereitä voi olla esimerkiksi löydettyjen virheiden määrä, määritellyn testausajan ja varatun rahamäärän loppuminen. Löydetyt virheet kirjataan virheraporttiin, johon voidaan kirjata esimerkiksi virheen kuvaus, vakavuus sekä löytökeino ja -ajankohta. (Haikala & Mikkonen 2011, 210–216)

3 OHJELMISTON SUUNNITTELU KOHDEYRITYKSELLE

LST Group Oy yhteistyössä TFT:n (Turku Future Technologies) kanssa hakivat yhteistyökumppania korkeakouluista resurssienhallinta-ohjelmiston kehittämiseen ja laativat tarvekartoituksen ohjelmiston tarpeellisista ominaisuuksista. Tehtävä päätyi Turun ammattikorkeakoululle. Tämä luku käsittelee ohjelmiston suunnitteluprosessin kulkua ja tuloksia.

Tarvekartoitusdokumentin perusteella järjestettiin sidosryhmien kesken useampi tapaaminen, missä kirjattuja vaatimuksia tarkennettiin. Toimeksiantajien kanssa päästiin sopimukseen siitä, että projekti aloitetaan selvitysvaiheella, missä luodaan ohjelmiston vaatimukset, luodaan käyttötapauksia ja etsitään alustavia toteutusvaihtoehtoja. Tavoite selvitysvaiheen valmistumiseen asetettiin kesän 2017 alkuun.

Selvitysvaiheessa tuotetun materiaalin perusteella suunniteltiin, että ohjelmiston prototyyppiä lähtisi toteuttamaan oppilaat osana opintojakson suoritusta, The FIRMAn avustuksella tai jonkin ohjelmointiyrityksen tuottamana. Toteutusprosessissa hyödynnetään ketteriä menetelmiä ja tuotettua toiminnallisuutta esitellään LST Groupille tasaisin väliajoin. Suunnitteluvaiheen lopussa prototyyppiä alkoi kehittämään Turun ammattikorkeakoulun innovaatioprojektiin osallistunut opiskelijaryhmä.

3.1 Kohdeyritys ja toimintaympäristö

LST Group on sähköalan konserni, jonka toimintaan lukeutuu muun muassa sähkösuunnittelua, kokonaistoimituksia ja erikoiskomponenttien valmistusta. Konsernin liikevaihto on 17 miljoonaa euroa vuodessa ja työllistää 160 henkilöä (LST 2017c). Emoyhtiön lisäksi konserniin kuuluu seitsemän tytäryhtiötä, jotka puolestaan jakautuvat kahdeksaan eri tuotantoyksikköön. Osa tuotantoyksiköistä jakautuu vielä erikseen osastoihin. LST Group Oy:n vanhin yritys on vuonna 1962 perustettu Laivasähkötyö Oy. Itse konserni on perustettu vasta yrityksessä vuonna 1988 tapahtuneen sukupolvenvaihdoksen johdosta. (LST 2017a; LST 2017b)

LST Groupin työt tapahtuvat projekteina, mitkä saattavat olla kestoltaan useita kuukausia. Jokaisella tytäryhtiöllä on omat projektinsa, joten koko konsernin tasolla projekteja on useita käynnissä samaan aikaan. Tämä tuo haasteita projektien resurssien

jakamiseen niin materiaalikustannuksissa kuin asentajien työtuntien jakamisessakin. Henkilöstötarpeet elävät pitkin projektien elinkaarta, joten resurssien jakaminen ja resurssitilanteen päivittäminen tasaisin väliajoin on tärkeää, jotta vältytään projektien turhilta seisahtumisilta.

LST Groupin alaisena toimivat sähköasentajat myydään tietyksi määräajaksi asiakkaan kanssa perustettuun projektiin. Sama asentaja saattaa olla osallisena useammassa projektissa kerrallaan. Asentajat työskentelevät projektista riippuen erilaisissa ympäristöissä, kuten telakoilla, laivoilla tai rautateillä.

3.2 Ohjelmistotarve

LST Groupin tarvekartoitusdokumentissa on eriteltynä kaksi pääasiallista tarvealuetta: tilauskanta ja resurssivaraus. Tilauskannan tarpeita ovat tilausten syöttäminen tilaussopimuksen mukaan, sekä tilauksen pohjalta aloitetun projektin talous- ja resurssibudjetin päivittäminen ja seuranta. Resurssivaraus pitää sisällään työntekijöihin ja projektien työtuntien täyttämiseen liittyviä toimintoja, kuten asentajien projektiin varaamista, saatavuuden ja osaamisten tarkastelua.

Suurin painoarvo ohjelmistolla on kuitenkin toimitusjohtajan ja projektipäälliköiden välisessä projektien ja niiden työtuntiresurssien hallinnassa. Tarkoituksena on saada käyttöön työkalu millä pystyy muokkaamaan ajattelumallia ja työtottumuksia siihen suuntaan, että projektien budjettien päivittäminen tapahtuisi jokaisen projektipäällikön toimesta. Ohjelmistoon kaivataan myös visualisointia projektien luomisen avuksi, jotta saadaan laajempi kuva kustannuksista ja resurssien kohdentamisesta. Ohjelmiston tulisi olla myös web-pohjainen ja mobiilisti käytettävissä.

3.2.1 Tilauskanta

Jokaiselle projektille annetaan ennuste sen vaatimien kustannusten määrästä, jotta voidaan myöhemmin tarkastella, kuinka hyvin se piti paikkansa. Näin voidaan myös varautua kuluihin paremmin tulevaisuudessa. Aktiivisia projekteja yritystä kohden saa olla budjetoituna maksimissaan kaksikymmentä kappaletta. Ellei erikseen muuta mainita, tulee kulut merkitä euroina.

Kirjattaviin kustannuksiin lukeutuvat

- materiaalikustannukset
- työkulut
- sivukulut (vakioprosentti saadaan Visman kautta)
- kulukorvaukset
- muut kulut
- kate.

Jokaiselle projektille on syytä ennustaa myös resurssibudjetti, eli kuinka monta työtuntia se arviolta tarvitsee maaliin saattamiseksi. Resurssibudjettiin kirjattavia asioita ovat

- projektin aloituspäivämäärä
- projektin lopetuspäivämäärä
- budjetoitavat työtunnit
- projektin vetovastuussa olevan henkilön antama valmiusarvio (prosenteissa)
- kuinka työtunnit jakautuvat viikoittain.

Kustannus- ja resurssiarvioiden täyttämisestä tulee lähteä sähköposti-ilmoitus toimitusjohtajalle. Ennusteita tulee päivittää projektin edetessä kuukauden välein, joten projektipäällikkö saa sähköpostiinsa muistutuksen tästä toimenpiteestä.

Projektin päätyttyä täytyy kirjata ylös lopputiedot, joihin lukeutuvat

- neliöt
- lopullinen työtuntien määrä
- kuluneet tunnit neliötä kohden
- kuluneet materiaalit neliötä kohden.

Lopputietojen täyttämisestä halutaan lähtevän sähköpostimuistutus, jotta täyttäminen myös tapahtuisi. Tämän jälkeen lähtee vuorostaan sähköpostikuittaus täyttämisestä eteenpäin toimitusjohtajalle.

3.2.2 Resurssivaraus

Resurssivarauksessa tarve keskittyy projekteihin resursoitavien työtuntien kohdentamiseen. Työntekijöitä LST Groupilla on noin 110 asentajaa ja 10 projektipäällikköä. Asentajat ovat eri tytäryhtiöissä kirjoilla, mutta tarpeen mukaan varattavissa muiden yritysten projekteihin, mikäli ovat vapaana. Asentajan vapautumisesta halutaan sähköposti-

ilmoitus toimitusjohtajalle kolme päivää ennen tapahtumaa. Asentajan tila pitää muuttua varatuksi kaikille projektipäälliköille samalla, jotta ennenaikaista projektien välistä siirtelyä ei tapahtuisi.

Asentajan saatavuuteen vaikuttavat projektikohtaisen varauksen lisäksi myös se, onko hän lomalla tai työkykyinen. Tästä syystä pitäisi asentajien saatavuutta pystyä tarkastelemaan yrityksittäin. Työntekijöistä olisi hyvä olla kirjattuna myös mahdollisia erityisosaamisen tietoja, jotta pystytään kohdentamaan asentajat mahdollisimman tehokkaasti ja hyödyllisesti sekä tiedetään, millaista osaamista on vapaana käytettävissä.

Työntekijälle lähtee sähköposti-ilmoitus tapahtumasta, kun hänet varataan projektiin. Jotta työntekijät tietävät kunakin päivänä, mihin projektiin ja missä paikassa hänen läsnäoloaan vaaditaan, olisi tarpeellista joko luoda ohjelmistoon kalenterinäkymä tätä varten, tai integroida kyseiset ominaisuudet johonkin valmiiseen kalenteripalveluun. Kalenterista näkyisi myös lomat ja pyhäpäivät.

Kun työntekijöitä asetetaan projektiin, tulee projektipäällikön päivittää budjettiennustetta, jottei resurssitilanne päädy nollatasoon ja projektia jouduta siitä syystä keskeyttämään. Budjetin päivittämisestä tulee sähköpostimuistutus viikon välein.

3.3 Nykyinen ratkaisu

Nykyinen ratkaisu LST Groupin projektien ja varausten hallintaan perustuu Excel-taulukon, Visman ja Trellon yhdistelmään. Yhdistelmä ajaa asiansa, mutta on työläs päivittää, ja melko hankalasti tulkittava. Excel-taulukosta näkee karkeasti numeeriset arvot projekteihin varatuista asentajatyötunneista viikkoa kohden, yritystasolla. Budjettitiedot tallentuvat Vismaan ja erilliset asentajavaraukset hoidetaan Trellon kautta, jossa projektipäälliköt voivat siirtää haluamansa työntekijät projektiinsa. Mikään ei kuitenkaan estä liikuttelemasta työntekijöitä sen jälkeen, kun he ovat jo kerran varattu, joten tilanne ei ole kovinkaan selkeä. Excel-taulukosta myös näkee, etteivät kaikki projektipäälliköt päivitä omaa tauluaan, joten on vaikea saada kunnollista kokonaiskuvaa.

LST Groupin työntekijöiden tiedot ovat tällä hetkellä tallennettuna Nepton-palvelussa, josta tietoja voisi mahdollisesti hakea kehitettävän ohjelmiston omaan tietokantaan ja toisin päin. LST Group hyödyntää projektinhallinnassa myös Power BI -järjestelmää, josta voisi työntekijöiden tietojen tapaan integroida tietoa projekteista ohjelmiston omaan tietokantaan ja toisin päin.

3.4 Vaatimusmäärittely

Ensimmäinen vaatimuskartoitus tapahtui LST Groupin ja TFT:n toimesta, joten määrittelyä ei tarvinnut aloittaa täysin tyhjältä pohjalta. Kartoitusdokumentissa esiintyviä vaatimuksia käytiin läpi useassa palaverissa LST Groupin ja ohjelmiston kehittäjätahon kanssa, missä vaatimuksia tarkennettiin ja priorisoitiin palautteen mukaisesti. Seuraavaksi käydään läpi tarvekartoituksen tuloksena syntyneiden vaatimusten tarkentumista projektin edetessä sekä ohjelmiston lopulliset ja tarkennetut vaatimukset käyttäjätarinamuodossa.

3.4.1 Kehitysprosessi ja vaatimusten tarkentuminen

Ohjelmiston suunnitteluvaihe käynnistettiin keväällä 2017 ja sen pohjana toimivat tarvekartoitusdokumentissa kirjatut vaatimukset. Vaatimusten tulkintaa vaikeutti suunnitteluryhmän jäsenen sairastuminen, sekä kommunikoinnin painottuminen sähköpostiviesteihin eri asuinpaikkakunnista johtuen. Kevään edetessä projektiin liittyi innovaatio-kurssin opiskelijaryhmä kehittämään ohjelmiston prototyyppiä. Ohjelmiston vaatimusten kirjaamiseen ja priorisoimiseen sekä kommunikointiin suunnittelutahon ja prototyypin luojien välillä käytettiin Trelloa ja WhatsAppia.

Ensimmäinen tapaaminen ohjelmiston suunnitteluryhmän, prototyypin tekijöiden ja LST Groupin edustajien välille sovittiin tapahtuvaksi toukokuun 2017 loppuun. Tähän mennessä oli luotuna alustavat vaatimukset ja käyttötapauksia, sekä niiden pohjalta ohjelmiston prototyypin ensimmäinen versio. Alustavat vaatimukset ja käyttötapaukset lähetettiin LST Groupille sähköpostitse huhtikuun lopussa. Tapaamisessa tuli ilmi, että ohjelmiston ominaisuuksissa olisi syytä keskittyä enemmän projektien resurssien pitkän ajan suunnitteluun. Prototyypissä suunnittelunäkymät eivät olleet vielä kunnolla sisällytettyinä, vaan tekeminen oli keskittynyt muun muassa kalenterinäkymään ja työntekijöiden varaamiseen.

LST Group oli ottanut tässä vaiheessa Trellon käyttöön asentajien varaamiseen, joten tämän ominaisuuden prioriteetti ohjelmistossa laski. Yrityksen edustajat toivoivat prototyyppiin taulukkoa projektien tuntimäärien karkeaa seurantaa varten sekä kalenteriin näkyville käytettävä tuntimäärä viikkotasolla. Työntekijöiden varaamisessa haluttiin saatavuustilan muuttuvan kaikille resurssia hallitseville samanaikaisesti, jotta asentajia ei

pystyisi siirtämään projektista kesken pois ilman vapautusta. Projektipäälliköiden ajattelumalliin haluttiin vaikuttaa siten, että projektien budjettiennusteen päivittäminen tehtäisiin pakolliseksi. Tämä tapahtuisi mahdollisesti siten, että ohjelmistoon tehtäisiin tasaisin väliajoin muistutus päivittämisestä sisäänkirjautumisen yhteydessä. Muistutusta pystyisi tiettyyn pisteeseen asti siirtämään, jotta ohjelmiston muiden toimintojen käyttäminen olisi mahdollista. Myöhästyneestä päivittämisestä lähtisi tieto sähköpostilla toimitusjohtajalle. Tapaamisessa selvisi myös Visma-järjestelmässä käytettävät tiedot, joita ovat

- palkat
- palkkojen sivukulut (prosentti, jonka arvo vaihtuu vuosittain)
- kulukorvaukset
- asumiskulut
- alihankinnat
- materiaalit
- suunnittelu (alihankinta).

Seuraava tapaaminen oli elokuussa. Tapaamisessa painotettiin projektien resurssien kohdistamisen suunnittelussa numeroarvojen käyttämistä työntekijöiden nimien sijaan. Työntekijöillä on yksilöivä työntekijänumero Nepton-palvelussa, joka haluttiin näkyviin ohjelmiston työntekijöitä koskevassa varauksessa. Kokonaistuntimäärien asettamiseen projektissa haluttiin S-käyrää, missä pääpaino tunneille olisi projektin keski- ja loppuvaiheessa, jotta vältyttäisiin toimettomilta viikoilta ylivaraamisen seurauksena. Yhden projektin keston tunnusluvuiksi annettiin 10 000 tuntia, 40 viikkoa ja 40 tuntia työntekijää kohden viikossa. Kaikista tärkeimpänä kehittämisen kohteena pidettiin sitä, että ohjelmistoon saadaan kokonaiskuvana diagrammi, missä kaikki käynnissä olevat projektit ja niiden arviolta vaaditut tunnit olisivat päällekkäin samassa asteikossa. Tämä näkymä on oleellinen tulevaisuuden tilanteen ennakoimisessa. Kulujen syöttämisen haluttiin tapahtuvan useampaan kenttään ja niistä päivittyisi kokonaiskulut samassa näkymässä.

Kolmas tapaaminen saatiin sovituksi lokakuulle 2017, ja on kirjoittamishetkellä viimeinen tapaaminen LST Groupin ja prototyypin sen hetkisen kehittäjäryhmän välillä. Tähän tapaamiseen mennessä ei ollut paljoa uutta kehitettynä prototyyppiin, mutta viime tapaamisessa tärkeimpänä pidettyä diagrammia oli työstetty visuaalisesti toimivaksi. Innovaatiokurssin opiskelijaryhmä oli tässä vaiheessa vuotta saanut kurssin suorittamiseen vaadittavat työtunnit täyteen, ja syksyn myötä muu koulunkäynti otti oman aikansa, joten prototyypin kehittäjät päättivät lopettaa kehitysprojektin työstämisen. Turun

ammattikorkeakoulun edustaja kertoi selvittävänsä prototyypin jatkokehitysmahdollisuuksia, sekä vaatimusmäärittely tehtäisiin loppuun osana käynnissä ollutta opinnäytetyötä.

3.4.2 Lopulliset vaatimukset

Tässä alaluvussa ovat alustavien vaatimusten, LST Groupin kanssa järjestettyjen tapaamisten sekä innovaatiokurssin opiskelijoiden kanssa käytyjen viestien pohjalta luodut ohjelmiston lopulliset vaatimukset. Vaatimukset on luokiteltu kolmeen eri prioriteettiluokkaan, joista numero 1 tarkoittaa toteutusarvoltaan korkeinta prioriteettia, ja numero 3 pienintä. Osasta vaatimuksia on luotuna taulukkomuodossa käyttötapauskuvaus. Kyseiset taulukot sijaitsevat kuvaamansa toiminnon käyttäjätarinan kanssa saman otsikon alla, lueteltujen vaatimusten jälkeen.

Prioriteetti 1: pakolliset

Projektipäällikkönä pystyn syöttämään uuden projektin tiedot tilaussopimuksen mukaan ja ohjelmisto tallentaa sen tiedot sekä muutokset budjettiin (Taulukko 1).

Projektipäällikkönä pystyn syöttämään ohjelmistoon karkean arvion projektikohtaisesta työtuntitarpeesta, jota pystyn päivittämään.

Projektipäällikkönä pystyn vertailemaan arvioitua työtuntimäärää varsinaiseen käytettyyn työtuntimäärään rinnakkain.

Projektipäällikkönä haluan nähdä valmiuskäyrän projektien etenemisestä, missä Y-akseli kuvaa valmiustasoa ja X-akseli viikkojen määrää, jotta näen arvion projektien etenemistahdista ja valmiuden edistymisen viikkotasolla.

Projektipäällikkönä pystyn seuraamaan projektikohtaista työtuntitarvetta ja yrityskohtaisia vapaita asentajia, jotta niiden suuntaaminen tarvittavaan projektiin on helpompaa.

Projektipäällikkönä näen kalenterista käytettävät työtunnit viikkotasolla ja käyrän, missä Y-akseli kuvaa arvioituja tunteja ja X-akseli viikkoja.

Projektipäällikkönä pystyn hyödyntämään työtuntien kohdentamisen suunnittelussa S-käyrää käyttävää taulukkoa, missä työtuntien pääpaino on projektien keski- ja loppuvaiheessa, jotta vältän liiallista resurssipanostusta projektin alkuvaiheessa.

Toimitusjohtajana pystyn hyödyntämään diagrammia, missä käynnissä olevat projektit arvioituine tunteineen ovat aseteltuna päällekkäin resurssien totaalikäytön hahmottamiseksi.

Taulukko 1. Tilauksen syöttäminen ohjelmistoon -käyttötapaus.

Nimi	Tilauksen syöttäminen ohjelmistoon
Versio	1.0
Suorittaja	Projektipäällikkö
Esiehdot	Käyttäjä on kirjautunut ohjelmistoon sisään tunnuksella, jolla on tarvittavat oikeudet.
Kuvaus	1. Käyttäjä navigoi tilausten syöttämiseen 2. Käyttäjä syöttää tilauksen perustiedot (yritys, hinta, kesto, kustannukset) niille varattuihin kenttiin 3. Käyttäjä arvioi tilauksen viikoittaisen työtuntimäärän 4. Käyttäjä tallentaa syötetyt tiedot [Poikkeus: budjetissa ei riittävästi resursseja verrattuna annettuihin määriin] [Poikkeus: kaikkia tarvittavia tietoja ei ole annettu] 5. Budjettimuutokset päivittyvät 6. Ohjelmisto lähettää sähköpostikuittauksen tilauksen syöttäjälle ja taloushallintoon
Poikkeukset	Poikkeus: budjetissa ei riittävästi resursseja verrattuna annettuihin määriin: ohjelmisto ilmoittaa käyttäjälle virheelliset tiedot muokkausta varten. Poikkeus: kaikkia tarvittavia tietoja ei ole annettu: ohjelmisto ilmoittaa käyttäjälle tarvittavien tietojen puuttumisesta.
Lopputulos	Tilaus on syötetty ohjelmistoon.
Muut vaatimukset	Ohjelmiston täytyy olla toimintakunnossa.

Prioriteetti 2: mahdolliset

Projektipäällikkönä olen velvoitettu päivittämään projektin budjettiennustetta työntekijää lisättäessä kirjautumisen yhteydessä, jotta työtuntien määrä ei päädy nollatasoon.

Projektipäällikkönä saan sähköpostimuistutuksen kerran viikossa budjettiennusteen päivittämisestä, joka täytyy kuitata ohjelmiston kautta.

Projektipäällikkönä voin viivästyttää budjettiennusteen päivittämistä viidellä minuutilla kirjautuessani ohjelmistoon, jotta pääsen tarvittaessa käsiksi ohjelmiston muihin ominaisuuksiin.

Toimitusjohtajana saan sähköposti-ilmoituksen budjettiennusteen päivittämisen myöhästymisestä.

Projektipäällikkönä pystyn varaamaan vapaan asentajan projektiin päiväkohtaisesti joko kokonaisen tai puolikkaan päivän ajaksi, sekä vapauttamaan tai siirtämään asentajan projektista toiseen tarpeen päätyttyä (Taulukko 2).

Projektipäällikkönä näen viikkonumerot, viikonloput, pyhäpäivät ja lomat tehdessäni päiväkohtaista asentajavarausta, jotta osaan valita sopivat varausajankohdat.

Projektipäällikkönä pystyn hyödyntämään dynaamista näkymää, josta selviää x-määrän valittujen työntekijöiden arviolta vaadittavat työtunnit viikkotasolla.

Projektipäällikkönä pystyn projektia luodessa hyödyntämään visualisointia, jossa eri kulut ovat eriteltyinä, sekä kaikkien kulujen summa näkyvillä.

Projektipäällikkönä en ole velvoitettu laskemaan itse sivukuluja, vaan ne on automatisoitu ennalta määrätyn prosentin mukaisesti.

Työntekijänä saan tiedon varauksestani sähköpostiini.

Toimitusjohtajana saan sähköpostiini tiedon kolme päivää ennen asentajavarauksen vapautumista.

Projektipäällikkönä saan tilannekatsauksen projektista sähköpostiini kerran kuukaudessa, jotta näen mahdolliset muutokset.

Tilauksen tallentaneena työntekijänä saan tallennuksen onnistumisesta varmistuksen sähköpostiini.

Taloushallinnon työntekijänä saan sähköposti-ilmoituksen projektin luomisesta ja päivittämisestä.

Taulukko 2. Asentajan lisääminen projektiin -käyttötapaus.

Nimi	Asentajan lisääminen projektiin
Versio	1.0
Suorittaja	Projektipäällikkö
Esiehdot	Käyttäjä on kirjautunut sisään ohjelmistoon tunnuksella, jolla on vaadittavat oikeudet. Projekti, johon resurssi lisätään, on luotuna.
Kuvaus	1. Käyttäjä navigoi projektien tarkasteluun 2. Käyttäjä valitsee kohdeprojektin 3. Käyttäjä navigoi asentajien lisäämiseen ja valitsee vapaana olevan asentajan [Poikkeus: vapaita asentajia ei ole] 4. Käyttäjä lisää valitun asentajan valittuun projektiin 5. Asentajan varaustila muuttuu kaikille tätä hallitseville tahoille 6. Ohjelmisto lähettää valitun asentajan sähköpostiin tiedon lisäyksestä
Poikkeukset	Poikkeus: vapaita asentajia ei ole: vapaita asentajia ei ole, joten lisääminen ei ole mahdollista ennen kuin asentajia vapautuu/vapautetaan tai muuten lisätään.
Lopputulos	Asentaja on lisätty projektiin.
Muut vaatimukset	Ohjelmiston täytyy olla toimintakunnossa.

Prioriteetti 3: muuta

Projektipäällikkönä näen työntekijän Nepton-palveluun tallennetun yksilöivän tunnusluvun asentajavarausta tehdessä.

Projektipäällikkönä täytän lopputiedot projektin päätyttyä ja tallennan ne, jotta niitä voidaan tarpeen tullen myöhemmin tarkastella (Taulukko 3).

Projektipäällikkönä voin tarkastaa esimerkiksi kaikki lomalla olevat työntekijät kerralla, jotta vapaan resurssin kohdentaminen helpottuu.

Projektipäällikkönä voin tarkastaa asentajien erityisominaisuuksia osaamismatrsiisista.

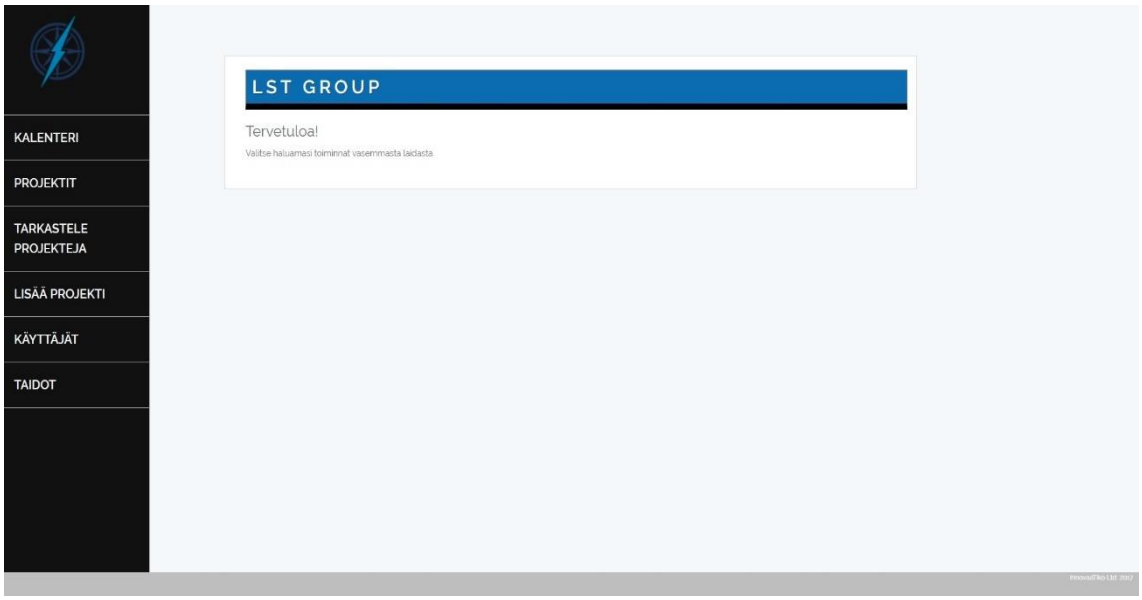
Taulukko 3. Projektin lopputietojen täyttäminen -käyttötapaus.

Nimi	Projektin lopputietojen täyttäminen
Versio	1.0
Suorittaja	Projektipäällikkö
Esiehdot	Käyttäjä on kirjautunut sisään ohjelmistoon tunnuksella, jolla on vaadittavat oikeudet. Tietojen täyttämisen kohteena oleva projekti on luotuna ja päättynyt.
Kuvaus	1. Käyttäjä navigoi projektien tarkasteluun 2. Käyttäjä valitsee kohdeprojektin 3. Käyttäjä navigoi projektin lopputietojen täyttämiseen 4. Käyttäjä syöttää tarvittavat lopputiedot 5. Käyttäjä tallentaa syötetyt lopputiedot [Poikkeus: kaikkea tarvittavaa lopputietoa ei ole syötetty] [Poikkeus: syötetty tieto ei kelpaa (esim. väärä formaatti)] 6. Ohjelmisto lähettää sähköpostikuittauksen lopputietojen täyttämisestä (kenelle?)
Poikkeukset	Poikkeus: kaikkea tarvittavaa lopputietoa ei ole syötetty: ohjelmisto ilmoittaa käyttäjälle, että kaikkia tarvittavia tietoja ei ole syötetty. Poikkeus: syötetty tieto ei kelpaa (esim. väärä formaatti): ohjelmisto ilmoittaa käyttäjälle, että syötetyssä tiedossa on virheitä.
Lopputulos	Projektin lopputiedot on täytetty.
Muut vaatimukset	Ohjelmiston täytyy olla toimintakunnossa.

3.5 Ohjelmiston prototyyppi

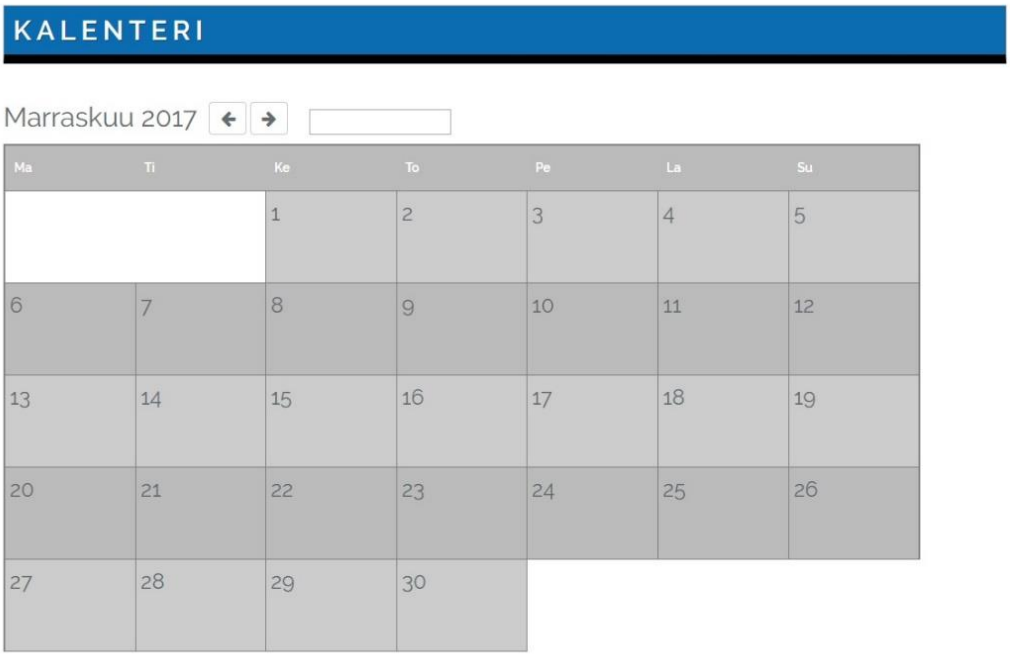
Innovaatiokurssin opiskelijat ehtivät toteuttamaan osallistumisensa aikana ohjelmistosta alustavan prototyypin, jossa on hieman visualisointia ja toimintoja. Kokonaisuudessaan se ei vielä täyttänyt vaatimuksia riittävällä tasolla, jotta se olisi kelvannut sellaisenaan käyttöön LST Groupille. Prototyyppi on web-pohjainen ja luotu Laravel-kehykseen keväällä 2017 kerättyjen vaatimusten ja käyttötapausten sekä kesän kehitystapaamisista saadun palautteen perusteella.

Aloitusnäkymässä on tervehdysteksti keskellä ja vasemmassa laidassa valikko, jonka kautta ohjelmiston eri osioissa navigoidaan (Kuva 2). LST Groupin logoa painamalla käyttäjä palaa aloitusnäkymään. Valikon voi piilottaa näkyvistä siinä tapauksessa, että sisältö ei näy ruudun koon takia kokonaisuudessaan.



Kuva 2. Prototyypin aloitusnäkymä.

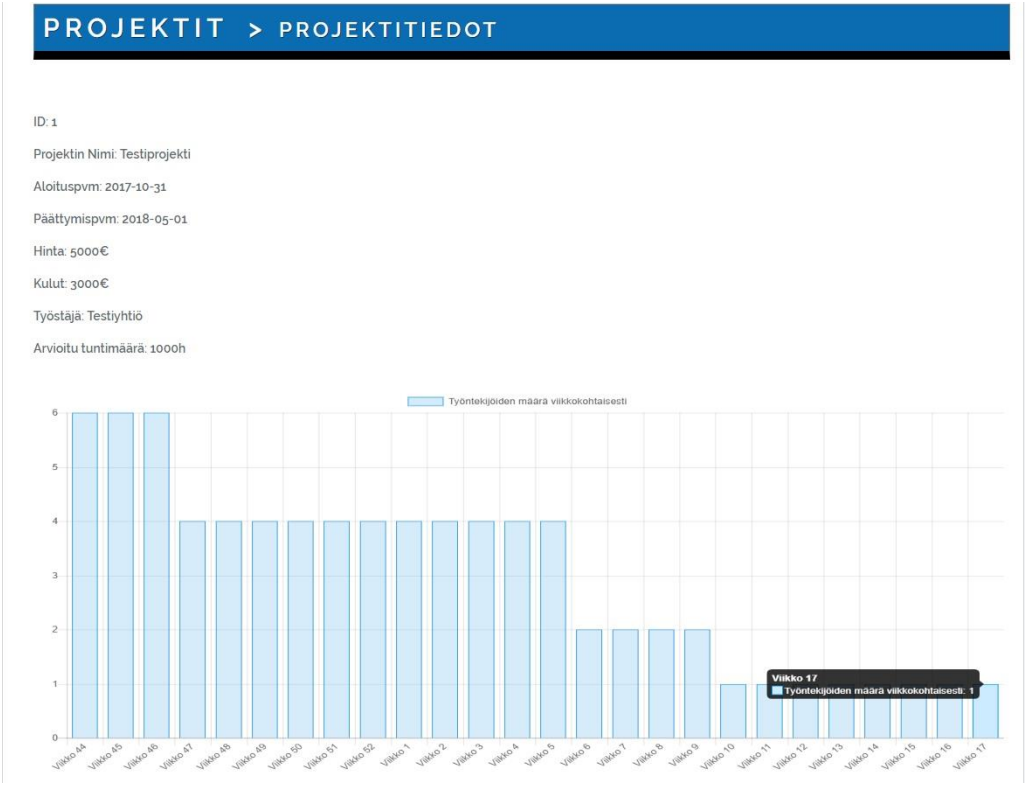
Kalenteri-osiossa on kalenterinäkymä, joka on visuaaliselta ulkoasultaan perinteinen (Kuva 3). Päivämäärälaatikkoa painamalla pääsee näkemään kunkin päivän varaukset, tosin tämä ominaisuus on keskeneräinen ja sisältää ainoastaan karkeaa tekstisisältöä, mikäli sitä on ohjelmistoon syötetty. Kuukausien välillä voi kulkea eteen- ja taaksepäin osoittavilla nuolipainikkeilla kalenterin vasemmassa yläkulmassa.



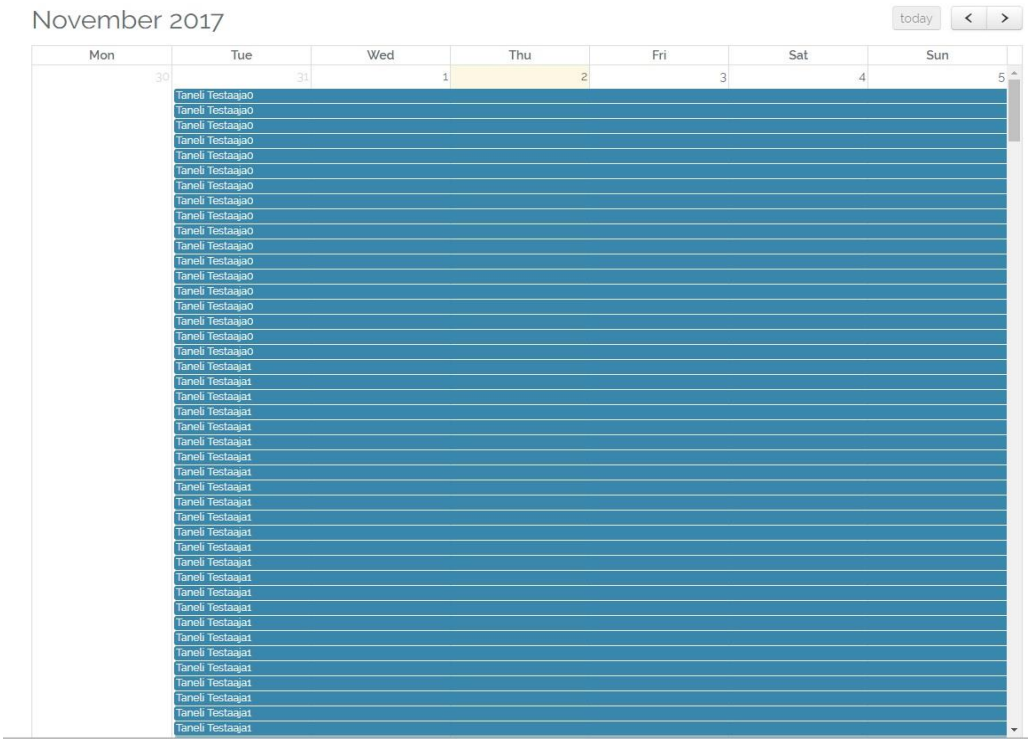
Kuva 3. Prototyypin kalenterinäkymä.

Projektit-osion takaa löytyy erikseen otsikoituna prototyypin sisäisesti luodut projektit, joita painamalla pääsee siirtymään projektin tietojen tarkasteluun (Kuva 4). Projekteista näkyvää tietoa on sen perustiedot liittyen nimeen, aloitus- ja lopetuspäivämäärään, budjettilukuihin ja työstäjään. Näkymästä selviää myös projektille asetettu työtuntimäärän kokonaisarvio sekä taulukko, jossa on arvio projektin viikoittaisesta jakaumasta työtuntimäärässä (Kuva 5). Taulukon alapuolella sijaitsee vielä kalenterinäkymä projektista, jossa on merkittynä testiasentajia heidän projektiin varattujen työtuntiensa mukaan (Kuva 6).

Kuva 4. Luodut projektit.

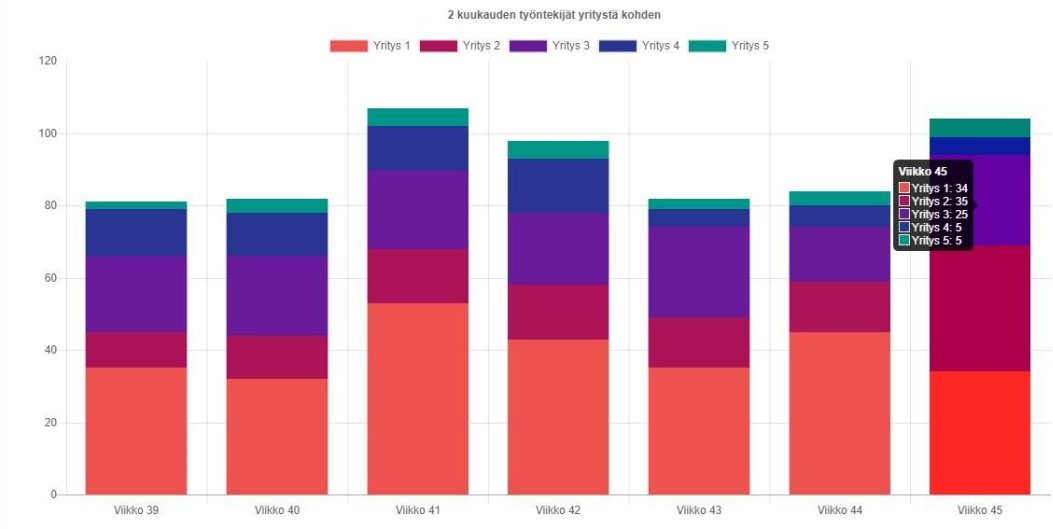


Kuva 5. Projektin perustiedot ja viikoittainen työtuntimäärän jakauma.



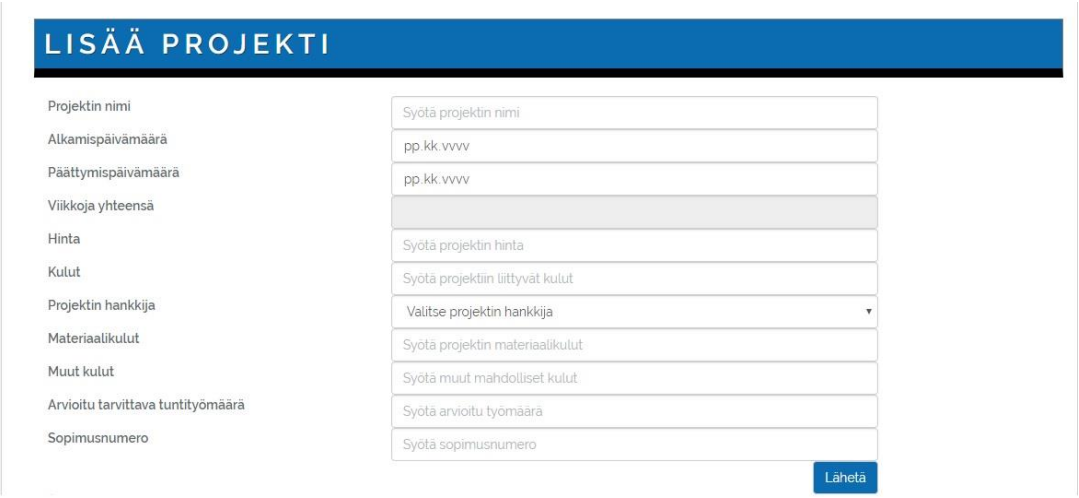
Kuva 6. Projektin kalenterinäkymä.

Tarkastele projekteja -osiossa on visuaalinen toteutus vaatimuksissakin kuvatusta diagrammista, jossa kaikki käynnissä olevat projektit ovat sijoitettuna samaan näkymään (Kuva 7). Viikkoa koskevat työntekijöiden lukumäärät saa numeerisesti näkyville liikuttamalla hiiren minkä tahansa diagrammin palkin päälle.



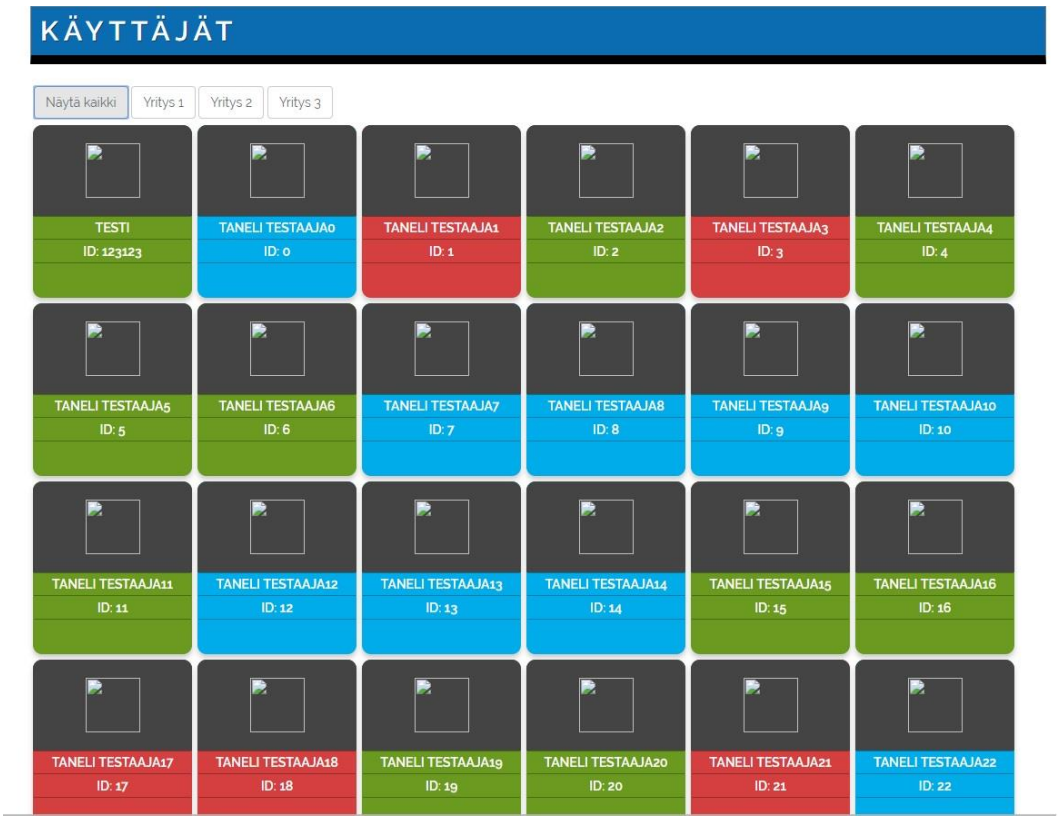
Kuva 7. Diagrammi käynnissä olevista projekteista.

Lisää projekti -näkymä on nimensä mukaisesti kehitetty projektien lisäämistä varten (Kuva 8). Projektin tiedot syötetään tilaussopimuksen mukaisesti. Projektin kesto viikkoina päivittyy automaattisesti syötetyn aloitus- ja lopetuspäivämäärän mukaan. Näkymän lopussa voi valita työntekijöitä projektiin alasvetovalikosta. Tietojen täyttämisen jälkeen ne tallennetaan painamalla lähetä-nappia.



Kuva 8. Lisää projekti -näkymä.

Viimeiset kaksi osiota liittyvät lähinnä asentajiin. Käyttäjät-osiossa on luotuna profiilikortteja työntekijöistä, joissa on kerrottuna nimi ja yksilöivä tunnus, sekä paikka kuvalle (Kuva 9). Vasemmassa yläkulmassa sijaitsevien valintanappien mukaan voi näkymää muokata näyttämään joko kaikki työntekijät kerralla tai jonkin yksittäisen yrityksen kirjoilla olevat työntekijät. Taidot-osio on luotu asentajien erityisosaamisten hakua varten, jossa haluttua erityisosaamista kuvaavaa otsikkoa painamalla saa listan kyseisen erityisosaamisen omaavista työntekijöistä (Kuva 10).



Kuva 9. Käyttäjät-näkymä.

Kuva 10. Taidot-näkymä.

4 YHTEENVETO

Opinnäytetyön tavoitteena oli tuottaa hyväksyttävä suunnitelma käyttäjälähtöisesti kohdeyrityksen tarpeita vastaavasta resurssienhallintaohjelmistosta, mihin sisältyy tarkennettu vaatimusmäärittely. Määrittelyn pohjalta toteutetaan ketterästi jonkinlainen rautalankamalli tai prototyyppi mahdollisen jatkokehityksen pohjaksi.

Kevään 2017 aikana luodut vaatimukset ja käyttötapaukset hyväksyttiin ja jatkokehitys alkoi. Ohjelmiston prototyypin luomiseen ilmoittautui opiskelijaryhmä, joka kehitti alustavien vaatimusten perusteella ensimmäisen version toukokuun loppuun sovittuun tapaamiseen. Tapaamisessa käydyn keskustelun perusteella vaatimuksia tarkennettiin ja priorisoitiin. Ohjelmiston prototyyppiä kehitettiin kesän ja syksyn aikana eteenpäin ja projekti tuli päätökseensä lokakuussa.

Ohjelmiston suunnitteluprojekti oli kiinnostava ja ajoittain todella haastava, sillä vaatimusten tulkitseminen kohdeyrityksen edustajien, vaatimusmäärittelyn luojan ja prototyypin kehitysryhmän välillä oli ajoittain toisistaan eroavaa. Tästä johtuen vaatimusten ja kehittämisen kohteiden priorisointia jouduttiin paikoitellen miettimään uudestaan. Vaatimuksissa nousi lopulta esiin vahvasti projektien resurssien käytön ja jakautumisen ennustaminen, sekä ennusteiden vertaaminen käytännössä toteutuneisiin arvoihin. Jotta näitä tietoja voidaan luotettavasti hyödyntää, tulee projektipäälliköt sitouttaa päivittämään niitä, ja ohjelmiston tulee olla riittävän hyvä käytettävyydeltään, jotta budjettien päivittäminen on helposti ja käytännöllisesti toteutettavissa.

Ohjelmiston käyttötapauksia saatiin luotua ja vaatimukset saatiin tarkennettua, priorisoitua ja muutettua käyttäjätarinoiksi. Prototyypin viimeisin versio ei sisällä kaikkea vaatimuksissa kuvattuja toimintoja ja ominaisuuksia, sillä kehitysprojektin aika loppui kesken. Suunnitelma oli onnistunut, sillä luotujen vaatimusten, käyttötapausten ja prototyyppikuvauksen perusteella voidaan ohjelmisto tulevaisuudessa kehittää.

LÄHTEET

Forsberg, K.; Mooz, H. & Cotterman, H. 2004. Projektin hallinta: malli kaupalliseen ja tekniseen menestykseen. Helsinki: Edita Publishing Oy.

Github 2017. About Laravel. Viitattu 4.11.2017 https://github.com/laravel/framework.

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. 12., uudistettu painos. Helsinki: Talentum Media Oy.

Huotari, S.; Laitakari-Svärd, I.; Laakko, J. & Koskinen, I. 2003. Käyttäjäkeskeinen tuotesuunnittelu. Käyttäjätiedon keruu, mallintaminen ja arviointi. Helsinki: Taideteollinen korkeakoulu.

Hyysalo, S. 2006. Käyttäjätieto ja käyttäjätutkimuksen menetelmät. 1. painos. Helsinki: Edita Publishing Oy.

JHS-suositukset 2012. ICT-palvelujen kehittäminen: Vaatimusmäärittely. Viitattu 14.10.2017 http://jhs-suositukset.fi/web/guest/jhs/recommendations/173.

JHS-suositukset 2016. Julkisten verkkopalvelujen suunnittelu ja kehittäminen. Viitattu 14.10.2017 http://docs.jhs-suositukset.fi/jhs-suositukset/JHS190/JHS190.pdf.

Joensuun yliopisto 2007. Integrointi- ja järjestelmätestaus. Viitattu 23.10.2017 http://cs.joensuu.fi/tSoft/testaus.htm.

Kettunen, S. 2009. Onnistu projektissa. 2., uudistettu painos. Helsinki: WSOYpro.

LST Group Oy 2017a. LST Group. Konsernin kuvaus. Viitattu 2.10.2017 http://www.lst.fi/lst-group/.

LST Group Oy 2017b. LST Group. Historia. Viitattu 2.10.2017 http://lst.fi/lst-group/historia/.

LST Group Oy 2017c. Rekrytointi. Avoimet työpaikat. Viitattu 4.10.2017 http://lst.fi/rekrytointi/avoimet-tyopaikat/.

Nepton 2017. Haasteet ja ratkaisut. Viitattu 23.10.2017 https://nepton.fi/haasteet-ja-ratkaisut/.

Nichols, K. & Chestnut, D.2014. UX For Dummies. Chichester: John Wiley & Sons, Ltd.

Nielsen, J. 1993. Usability Engineering. 1. painos. San Francisco: Morgan Kaufmann.

Opetushallitus 2017. Suomenkielinen sanasto. Viitattu 23.10.2017 http://www.oph.fi/tietopalvelut/ennakointi/koulutus_ja_osaamistarpeiden_ennakointi/sanastot/suomenkielinen_sanasto#S.

Power BI 2017. Tour. Viitattu 23.10.2017 https://powerbi.microsoft.com/en-us/tour/.

Projekti-instituutti 2017. Projektiohjelmiston käyttöönotto. Viitattu 10.11.2017 https://www.projekti-instituutti.fi/palvelut/ohjelmistokonsultointi/projektiohjelmiston_kayttoonotto.

Sinkkonen, I.; Nuutila. E. & Törmä, S. 2009. Helppokäyttöisen verkkopalvelun suunnittelu. Helsinki: Tietosanoma Oy.

Sourcemaking 2017. Use Case Diagrams. Viitattu 23.10.2017 https://sourcemaking.com/uml/modeling-business-systems/external-view/use-case-diagrams.

Techterms 2017a. Excel. Viitattu 23.10.2017 https://techterms.com/definition/excel.

Techterms 2017b. UML. Viitattu 23.10.2017 https://techterms.com/definition/uml.

The FIRMA 2017 The FIRMA. Viitattu 28.10.2017 http://thefirma.fi/.

Tolvanen, P. 2013. Ketteryys haltuun: Yleisimmät ketterät käytännöt. Sininen Meteoriitti. Yritys. Viitattu 7.10.2017 https://www.meteoriitti.com/2013/06/06/ketteryys-haltuun-yleisimmat-ketterat-kaytannot/.

Trello 2017. Tour. Viitattu 23.10.2017 https://trello.com/tour.

Visma 2017. Yksi kumppani. Viitattu 23.10.2017 https://www.visma.fi/tietoa-vismasta/yksi-kumppani-kaikki-palvelut-visma/.

WhatsApp 2017. Tietoja WhatsAppista. Viitattu 4.11.2017 https://www.whatsapp.com/about/.