



ASIAKASPROJEKTIN MÄÄRITTELYPROSESSIN JA -DOKUMENTAATION KEHITTÄMINEN

Viivi Hildén

Opinnäytetyö
Joulukuu 2011
Tietojenkäsittelyn koulutusohjelma
Ohjelmistotuotanto
Tampereen ammattikorkeakoulu

TAMPEREEN AMMATTIKORKEAKOULU

Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Ohjelmistotuotannon suuntautumisvaihtoehto

HILDÉN, VIIVI: Asiakasprojektin määrittelyprosessin ja -dokumentaation kehittäminen

Opinnäytetyö 46 sivua, liitteet 30 sivua.
Joulukuu 2011

Tässä opinnäytetyössä perehdytään ohjelmistotuotannon määrittelyprosessiin ja -dokumentaatioon. Työn tavoitteena oli luoda työn tilaajalle kustannustehokkaampi ja hallittavampi määrittelyprosessi henkilöstöhallinnan työkalujen asiakasprojekteihin. Työn tarkoituksena oli selvittää nykyisen määrittelyprosessin ongelmakohtat ja pyrkiä ratkaisemaan ne sopivilla kehitystoimilla.

Työn kirjallisessa osuudessa käsitellään ohjelmistomäärittelyä teoriassa ja selvitetään toimeksiantajan ohjelmistomäärittelyn nyky- sekä tavoitetilaa. Selvitettyjen nykytilan ongelmakohtien perusteella tehtiin kehitysratkaisuja, jotka yhdessä muodostivat työn tulokset.

Työn tuloksina syntyi toimeksiantajan käyttöön pieni joukko asiakirjapohjia ja ohjeistus määrittelyyn. Laaditut asiakirjapohjat ovat konseptisuunnitelma, käytötapaaukset ja vaatimusmäärittely. Määrittelyn ohjeistus sisältää ohjeet dokumenttien laadintaan, käsittelyyn, ylläpitoon, versionhallintaan sekä määrittelyiden purkamiseen tiketeiksi. Työn teoriaosuutta ja määrittelyn ohjeistusta voidaan käyttää osana perehdytysmateriaalia.

Kehitystoiminnan tuloksina määrittelyprosessista ja -dokumentaatiosta saatiin luotua selkeämpi ja käytännöllisempi kokonaisuus. Työlle asetetut tavoitteet saavutettiin ja työn tuloksia on alettu hyödyntää työn tilaajan liiketoiminnassa.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information System
Option of Software Engineering

HILDÉN, VIIVI: Improving the Requirements Process and Documentation in Customer Projects

Bachelor's thesis 46 pages, appendices 30 pages
December 2011

The objective of this thesis work was to create the more cost effective and manageable requirements process for the client. The purpose of this thesis was to discover problems in the current requirements process and find suitable ways to improve it.

The report includes a theoretical section, current state analysis, analysis of needs, development action and results. Based on the analyses that were made a series of development actions were decided upon. The results from these development actions form the results of the thesis.

The results include requirements templates and a guide to requirements process for the client. The guide functions as an introduction to good policies for the clients' requirements process and documentation. The results of this thesis have been utilized in the everyday work of the clients' employees.

Key words: Software engineering, requirements, documentation

SISÄLTÖ

TIIVISTELMÄ	2
ABSTRACT	3
1 JOHDANTO	5
1.1 Toimeksiantaja, tavoite ja tarkoitus	5
1.2 Aiheen rajaukset	6
2 OHJEMISTOMÄÄRITTELY	7
2.1 Mitä määrittely on?	7
2.2 Miksi määritellä?	8
2.3 Määrittelyn haasteet	10
2.4 Mitä määritellään?	12
2.4.1 Toiminnalliset vaatimukset	14
2.4.2 Ei-toiminnalliset vaatimukset	15
2.5 Määrittelyn toimijat	16
2.6 Määrittelyn vaiheet	19
2.6.1 Projektin aloitus	21
2.6.2 Vaatimusten keräys	21
2.6.3 Vaatimusten prototyypit	23
2.6.4 Vaatimusten kirjoittaminen	24
2.6.5 Vaatimusten uudelleenkäyttö	25
2.6.6 Vaatimusten validointi	25
2.6.7 Dokumentaation ylläpito	26
3 NYKYTILA	27
3.1 Määrittelyn nykytila toimeksiantajalla	27
3.2 SWOT- analyysi	28
3.2.1 Vahvuudet	29
3.2.2 Heikkoudet	30
3.2.3 Mahdollisuudet	31
3.2.4 Uhat	31
3.3 Ongelmakohtat	32
4 TAVOITETILA	34
4.1 Kehityskohteet	34
4.2 Vaatimukset	35
4.3 Ongelmakohtat ja rajoitteet	36
5 KEHITTÄMISPROSESSI	37
5.1 Toteutettavat asiakirjapohjat	37
5.2 Toteutettava ohjeistus	37
5.3 Tavoitetilan saavuttaminen	38
6 KEHITTÄMISTOIMINNAN TULOKSET	41
6.1 Opinnäytetyön eteneminen	41
6.2 Työn teoriaosuus	42
6.3 Toteutetut asiakirjapohjat	42
6.4 Toteutettu ohjeistus	43
7 POHDINTA	44
LÄHTEET	46
LIITTEET	47

1 JOHDANTO

Ohjelmistoyritysten määrä, ohjelmistojen koko ja asiakkaiden vaatimukset ovat kasvaneet viimeisien vuosien aikana räjähdysmäisesti. Teknologian kehityksen myötä pystytään rakentamaan aina parempia ja vaativampia ohjelmistoja. Ohjelmistojen määrittely ja dokumentointi on kuitenkin suuressa osassa yrityksiä jäänyt kehityksestä jälkeen. Ohjelmistojen määrittelyprosessi ja -dokumentaatio eivät usein vastaa ohjelmilta vaadittua tasoa. Tämän opinnäytetyön tarkoituksena on nostaa työn toimeksiantajan henkilöstöhallinnan työkalujen määrittelyprosessi ja -dokumentaatio yrityksen tuotteiden laadun tasolle.

Opinnäytetyössä perehdytään ohjelmistomäärittelyn prosessina. Prosessi sisältää vaatimusten löytämisen, dokumentoinnin ja ylläpidon. Ohjelmistomäärittely ja määrittelydokumentaatio ovat tärkeimpiä osia ohjelmistoprojekteissa, ja usein myös laiminlyödyimmät osat. Tarve kehittää työn toimeksiantajan asiakasprojektien määrittelydokumentaatiota ilmeni selvästi heti työharjoitteluni alettua testaaja/määrittelijänä yrityksen palveluksessa. Työn tavoitteena ei ole vakiinnuttaa tarkkoja toimintamalleja, vaan tarjota työkaluja määrittelyn ja dokumentoinnin helpottamiseksi.

1.1 Toimeksiantaja, tavoite ja tarkoitus

Opinnäytetyön toimeksiantaja toteuttaa avoimeen teknologiaan pohjautuvia henkilöstöhallinnan työkaluja, sekä oppimisympäristöjä kuten Moodle. Opinnäytetyöni toteutetaan tukemaan henkilöstöhallinnan työkaluja. Työkalut ovat selainpohjaisia sovelluksia henkilöstön ja organisaation osaamisen hallintaan, kehittämiseen ja analysointiin.

Työn toimeksiantaja on toiminut jo pitkään henkilöstöhallinnan työkalujen asiantuntijana, mutta työkalujen kehittyttyä on ohjelmistojen vaatimusten dokumentointi jäänyt jälkeen. Henkilöstöhallinnan työkalujen puolella asiakasprojektien vaatimusmäärittelyn dokumentaatio on puutteellista, ja sen tilaan toivotaan pa-

rannusta. Opinnäytetyön tavoitteena on luoda asiakasprojektien määrittelydokumentaatiosta hallittavampi ja kustannustehokkaampi kokonaisuus. Opinnäytetyön tarkoituksena on selvittää nykyisen määrittelyprosessin ja -dokumentaation ongelmakohtia sekä pyrkiä etsimään ongelmakohtiin ratkaisuja. Määrittelydokumentaation kehittäminen on toimeksiantajalle ajankohtainen asia ja onnistuessaan opinnäytetyö edistää kehitystä merkittävästi.

Työ aloitetaan käymällä teoriassa läpi määrittelyprosessia. Teorian on tarkoitus selkeyttää määrittelyn eri vaiheita ja auttaa ymmärtämään työssä tehtyä kehitystyötä paremmin. Teoriaosuuden jälkeen kartoitetaan työn toimeksiantajan määrittelyprosessin ja -dokumentaation nykytila sekä ongelmakohtat. Nykytilan ja ongelmakohtien pohjalta rakennetaan toimeksiantajan toivoma tavoitetila. Kun tarpeet on selvitetty, päätetään mitä kehitystoimia työssä tullaan tekemään. Kehitystoiminnan tuloksina syntyneet asiakirjapohjat ja määrittelyn ohjeistus löytyvät opinnäytetyön liitteistä. Lopuksi tarkastellaan työn tuloksia ja onnistumista.

1.2 Aiheen rajaukset

Aihe rajataan käsittämään ainoastaan henkilöstöhallinnan asiakasprojektien määrittelyprosessin ja -dokumentaation kehittämistä. Työssä ei oteta kantaa asiakasprojektien tekniseen dokumentaatioon. Työ ei ota kantaa jo olemassa olevien asiakasprojektien dokumentaatioon. Dokumentaatiota ei aleta parantamaan takautuvasti jo toteutettuihin asiakasprojekteihin.

2 OHJEMISTOMÄÄRITTELY

Tässä luvussa käydään teoriassa läpi määrittelyä, sen tarkoitusta, tärkeyttä, haasteita sekä itse määrittelyprosessia ja sen toimijoita. Tarkoitus ei ole selvittää yksityiskohtaisesti jokaista tunnistettua vaihetta määrittelyprosessista, vaan luvuissa käydään läpi käytännöllisimpiä ja tärkeimpiä määrittelyn vaiheita. Tämä luku luo pohjan myöhemmin tehtäville työn toimeksiantajan määrittelyprosessin kehitystoimille, kuten nykytila-analyysille, tavoitetilan luomiselle ja kehitystoiminnalle.

2.1 Mitä määrittely on?

Yksinkertaista määrittelyä esiintyy jokapäiväisessä elämässä. Kun tarvitaan jotain kaupasta kirjoitetaan ostoslista, kun tarvitaan raportti myynneistä työkaverilta kirjoitetaan sähköposti, jossa selitetään mitä lukuja halutaan ja miltä ajalta. Ostoslista ja sähköposti ovat yksinkertaisia määrittelydokumentteja. Määrittelydokumentteja luodaan tilanteessa, jossa tarpeet toteuttamaan pystyväälle henkilölle kuvataan, mitä tarvitaan. (Rinzler 2009, 2.) Ohjelmistomäärittelyitä kerätään ja dokumentoidaan, jotta pystytään toteuttamaan asiakkaan toiveita vastaava tuote.

Jotta voi rakentaa ohjelmiston täytyy tietää, mitä on tarkoitus rakentaa. Jotta tietää, mitä on tarkoitus rakentaa, täytyy kartoittaa ja kerätä vaatimukset, eli määritellä tuote. Oli sitten kyse uudesta asiakkaalle rakennettavasta ohjelmasta tai olemassa olevien komponenttien avulla rakennettavasta ohjelmasta, täytyy ohjelma aina määritellä ennen kuin sen voi rakentaa.

2.2 Miksi määrittellä?

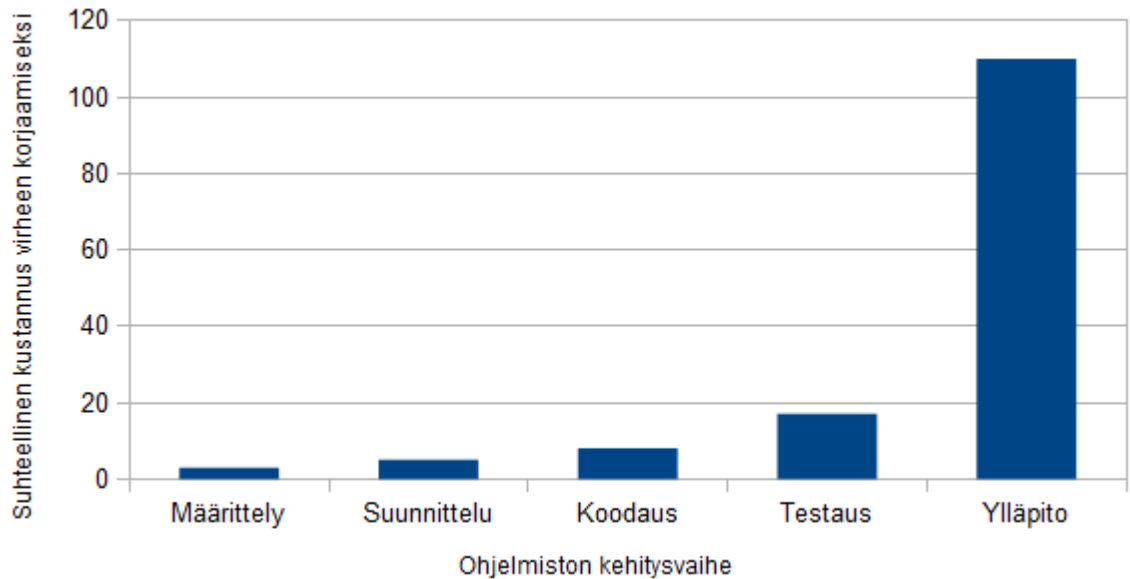
Miksi määrittelyä sitten tehdään? Määrittelyiden tarkoituksena on näyttää mitä tuloksia projektin toimijat vaativat, antaa toimijoille mahdollisuus sanoa mitä he haluavat, edustaa eri näkökulmia, tarkistaa suunnitelma, mitata kehitystä ja tarjota hyväksymiskriteerit tuotteen hyväksynnälle (Alexander & Stevens 2002, 15–16). Määrittelyn avulla pyritään varmistamaan että asiakas saa haluamansa ja ajallaan.

Ohjelmistomäärittelyn tarkoitus on saada tuotettua haluttu lopputulos aikataulun ja budjetin puitteissa. Huono vaatimusten keräys ja määrittely aiheuttaa ohjelmistovirheitä, jotka aiheuttavat ongelmia projektin etenemisessä. Ohjelmistovirheet aiheuttavat ylimääräistä työtä, kustannuksia ja aikataulun ylityksiä. Vain kun tietää oikeat vaatimukset, voi suunnitella ja rakentaa tuotteen, joka mahdollistaa tuotteen käyttäjien työskentelyn tavalla, joka tyydyttää liiketoiminnan vaatimukset (J. Robertson & S. Robertson 2010, 8). Kun projektin vaatimukset ovat selvillä, sujuu itse tuotteen rakentaminen sulavasti.

Projektin sujuvuuden takaamiseksi on tärkeää käyttää aikaa vaatimusten selvittämiseen ja niiden dokumentointiin. Vaikka määrittely tuntuisi ylimääräiseltä kustannukselta ja ajan tuhlaukselta, on totuus kuitenkin toinen. Hyvän määrittelyn saavuttamiseksi kuluu toki aikaa ja resursseja, mutta todellisuudessa se tuottaa lopulta säästöä koko projektin elinkaaren aikana. Lukuisissa tutkimuksissa on todettu huonosta määrittelystä aiheutuvan huomaita kustannuksia.

Uudelleen työstäminen voi kuluttaa jopa 30-50 prosenttia projektin kehityskustannuksista (Boehm & Papaccio 1988, 1462–1477, Wiegers 2003, 17 mukaan) ja määrittelyn virheet aiheuttavat 70–85 prosenttia uudelleen työstämisen kustannuksista (Leffingwell 1997, Wiegers 2003, 17 mukaan). Kuten kuviosta 1 selviää, projektin myöhäisessä vaiheessa löydetyn virheen korjaaminen maksaa suhteessa paljon enemmän, kuin virheen korjaaminen

nopeasti sen tekemisen jälkeen (Grady 1999, Wiegers 2003, 17 mukaan). Vaikka määrittelyvaiheessa tehdyt virheet eivät aiheuta kaikkia uudelleen työstämisen kustannuksia, on selvää, että jo pienienkin virheiden vähenemisen ansiosta korjauskustannukset voivat laskea selvästi.



KUVIO 1. Suhteellinen kustannus virheen korjaamiseksi (Grady 1999, muokattu)

Useimmiten huonosta määrittelystä johtuvat ongelmat ovat kustannuksiin ja aikatauluihin liittyviä, mutta joskus vaikutukset voivat olla laajemmat ja lähes katastrofaaliset. Ääripään esimerkkinä voidaan pitää 25 helmikuuta 1991 Saudi Arabiassa tapahtunutta vakavaa välikohtausta. Patriot ilmatorjunta ohjuspat-
 terin oli määrä suojella sotilastukikohtaa Dahrhnessa Saudi Arabiassa, mutta se epäonnistui tuhoamaan lähestyvän Scud ohjuksen. Epäonnistumisesta syytettiin virhettä ohjelmiston laskukaavoissa. Patriot hallinnointijärjestelmä piti kirjaa ajasta sekunnin kymmenesosien avulla; muuntaakseen luvun sekunneiksi kone kertoi ne luvulla 1/10. Matemaattisesti kiistämätön toiminto, mutta laskennallisesti katastrofaalinen. Koska laskennan 1/10 tuloksella ei ole tarkkaa esitystapaa binäärimerkistötavalla, kone joutui käyttämään likiarvoja. Ilmeisesti

muunnoksen vakio koneessa oli 24 binäärinen murtoluku 0.00011001100110011001100, joka on noin yhden miljoonasosan verran liian pieni. Eroavuus kuullostaa pieneltä, mutta neljän päivän aikana ero kasvoi sekunnin kolmannekseen. Yhdessä muiden hallinnointijärjestelmän erikoispiirteiden kanssa se aiheutti lähes 700 metrin laskuvirheen lähestyvän ohjuksen sijainnista. 28 sotilasta kuoli. (Hayes 2003, 484.) Sitä olisiko tarkempi määrittely ja testaus voinut pelastaa 28 henkeä on mahdoton jälkikäteen sanoa.

2.3 Määrittelyn haasteet

Puolet ohjelmistoprojekteista ylittävät joko budjetin tai aikataulun, usein molemmat. Useimmat ongelmat ohjelmistoprojekteissa johtuvat huonosta määrittelystä. Huonolla määrittelyllä tarkoitetaan, että jossain kohtaa määrittelyprosessia, tarpeita kerätessä, tietoa dokumentoidessa, validoinnissa tai tiedon muokkaamisessa on epäonnistuttu. Huono määrittelydokumentaatio ja sen ylläpito aiheuttaa ylimääräisiä kustannuksia, budjetin ylityksiä, epärealistisia odotuksia asiakkailta ja kesken jääneitä tappiollisia projekteja.

Haasteena on, että määrittelyä ei usein oteta vakavasti. Vaikka yrityksellä olisikin osaavia määrittelijöitä, ei projekteissa varata tarpeeksi aikaa ja resursseja huolelliseen määrittelyyn. Yrityksen johdon saattaa olla vaikea malttaa mielensä aikataulutuksessa, koska projektit halutaan saada nopeasti valmiiksi ja asiakkaan rahat omaan kassaan. Niinpä projektin aloitusta kiirehditään ja hypätään liian nopeasti tuotteen rakentamisvaiheeseen.

Johtoa voi olla hankala vakuuttaa määrittelyn tärkeydestä, mutta esimerkiksi seuraavista tuloksiin pohjautuvista perusteluista saattaa olla hyötyä. Vaikka projektin onnistumiseen tai epäonnistumiseen liittyy monia tekijöitä, ovat määrittelystä aiheutuvat virheet usein syynä kohtuuttoman suureen osaan virheitä. Ohjelmistokehityksessä 50–60 prosenttia virheistä juontaa juurensa vaatimusmäärittely- ja suunnitteluvaiheeseen. Näistä virheistä noin 5 prosenttia

juontaa juurensa kehityksen ohjelmointivaiheeseen. (J. Robertson & S. Robertson 2010, 262.) Viittä prosenttia lukuun ottamatta nämä virheet juontavat siis juurensa määrittelyvaiheessa tehtyihin virheisiin. Mikäli määrittely on puutteellista tai virheellistä, aiheutuu siitä korjauskustannuksia.

Jälkikäteen tehdyt korjaukset eivät ainoastaan tuhlaa rahaa, vaan myös aikaa. Eräs tutkimus totesi, että määrittelyvaiheessa löytyneen virheen korjaaminen vei keskimäärin 30 minuuttia, kun taas testausvaiheessa löydetty virheet ja bugit veivät keskimäärin viidestä (5) seitsemääntoista (17) tuntia (Kelly, Sherif & Hops 1992, Wiegers 2003, 260 mukaan). Huolellisella määrittelyllä ja määrittelyiden testaamisella jo niiden kirjoittamisvaiheessa voidaan säästää huomattavasti aikaa.

Vaikka yrityksen johto olisi saatu vakuutettua määrittelyn tarpeellisuudesta, on projektilla edessään vielä useita haasteita. Tarpeiden määrittely on hankala prosessi täynnä sudenkuoppia. Lisäksi suurimmalla osalla ihmisiä, jotka työskentelevät määrittelyn parissa, ei ole siihen valmistavaa koulutusta. Toisaalta määrittely vaatii teknistä ymmärrystä ohjelmistoista, jotta tietää mitä voidaan rakentaa ja mitä ei, mutta myös hyviä ihmissuhdetaitoja. Vaikka olisi teknisesti lahjakas, huonot vuorovaikutustaidot johtavat väistämättä vajavaisiin määrittelyihin.

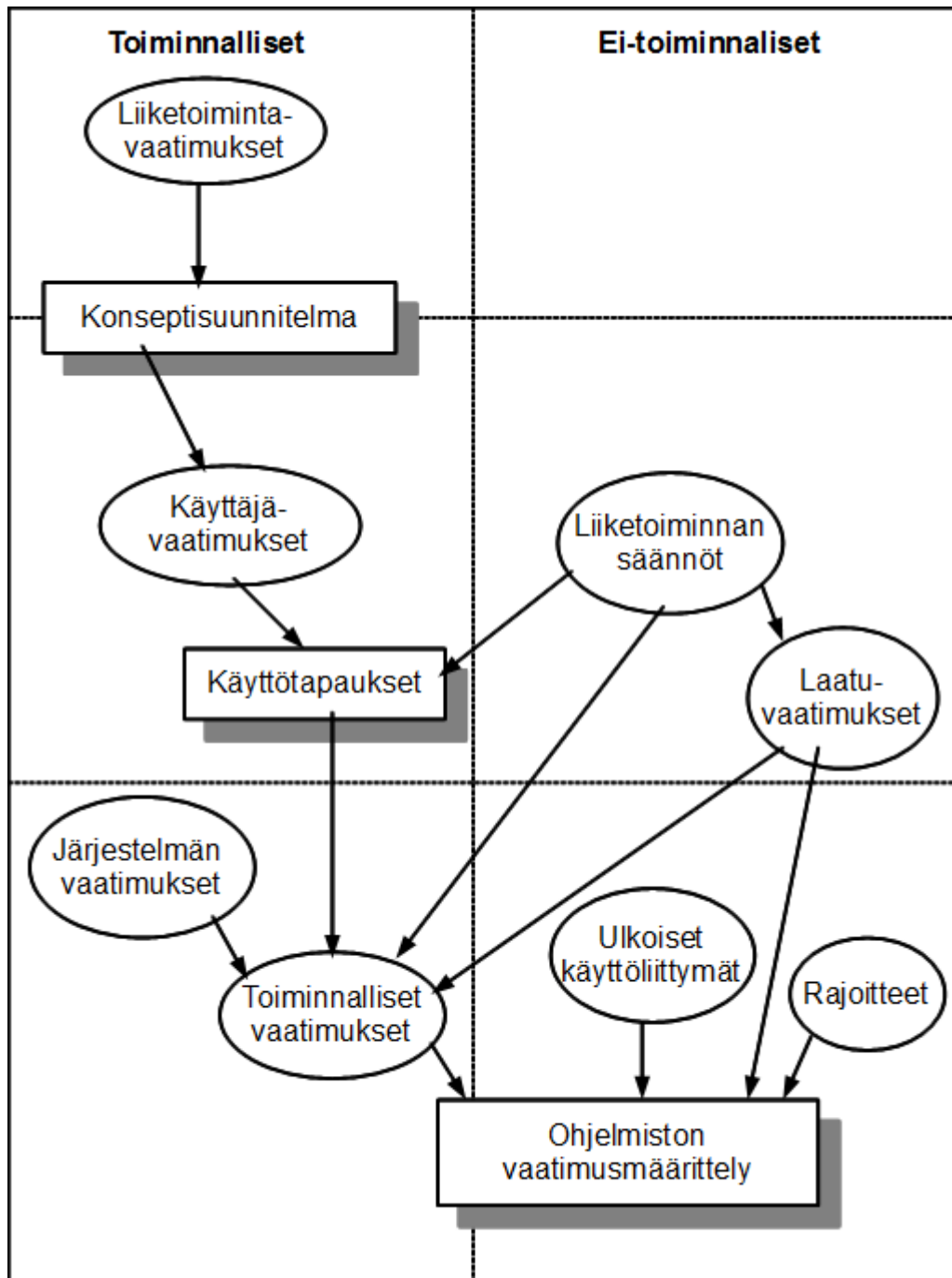
Tarpeiden kartoittamisessa täytyy huomioida, että usein asiakkaalla, joka tulee olemaan tuotteen loppukäyttäjä, ei ole ymmärrystä projektin teknisestä puolesta ja mahdollisista rajoitteista. Jokin mitä asiakas pitää oletuksena, eikä mainitsemisen arvoisena ominaisuutena, saattaa aiheuttaa ongelmia projektin myöhäisemmässä vaiheessa. Asiakkaiden kanssa on siis hyvä kirjoittaa kaikki pienimmätkin asiat ylös, vaikka ne saattavat tuntua itsestään selvyyksiltä. Hyvään määrittelyprosessiin ei ole mitään oikotietä. Ainoastaan ryhmä hyviä käytänteitä ja ohjeistuksia, joita jokainen voi soveltaa sopivammalla tavalla.

2.4 Mitä määritellään?

Ohjelmistomäärittelyssä pyritään keräämään asiakkaan vaatimukset ja tarpeet ohjelmistoon liittyen. Vaatimusten selvittäminen ei ole yksinkertaista ja itse vaatimuksista on olemassa monia eri määrittelyjä ja ajattelutapoja niiden luonnetta ja sisältöä ajatellen. IEEE Standard Glossary of Software Engineering Terminology (1990) määrittelee vaatimuksen olevan (Wiegiers 2003, 7):

1. Käyttäjän tarvitsema tila tai kyky ratkaista ongelma tai saavuttaa tavoite.
2. Järjestelmän tai järjestelmän osan tila tai kyky, joka vastaa sopimusta, standardia, määrittelyä tai muuta virallisesti määriteltyä dokumenttia.
3. Dokumentoitu vaste tilasta tai kyvystä kuten kohdassa 1 tai 2.

Vaatimukset jaetaan yleisesti toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Erilaisia toiminnallisia vaatimuksia ovat liiketoiminnan vaatimukset, käyttäjävaatimukset ja järjestelmän vaatimukset. Ei-toiminnallisia vaatimuksia ovat liiketoiminnan säännöt, laatuvaatimukset, ulkoiset käyttöliittymät ja rajoitukset. Kuvio 2 esittää eri vaatimusten välisiä suhteita. Kuvassa ovaalit ovat eri tyyppisiä vaatimuksia ja neliöt esittävät erilaisia säiliöitä tiedolle ja vaatimuksille.



KUVIO 2. Erityyppisten vaatimusten suhteet (Wiegiers 2003, 9, muokattu)

Kuten kuviosta selviää toiminnallisiin vaatimuksiin kuuluvat liiketoiminnan vaatimukset, käyttäjävaatimukset, järjestelmän vaatimukset, sekä toiminnalliset vaatimukset. Ei-toiminnallisten vaatimusten puolelle taas kuuluvat liiketoiminnan säännöt, laatuvaatimukset, ulkoiset käyttöliittymät ja rajoitteet. Seuraavissa alaluvuissa tarkastellaan lyhyesti näitä vaatimuksia ja niistä muodostuvia dokumentteja.

2.4.1 Toiminnalliset vaatimukset

Liiketoimintavaatimukset ovat yrityksen toiminnalleen asettamia tavoitteita. Liiketoiminnan tavoitteet selvitetään projektin konseptisuunnitelmaa varten. Vaatimuksena voi olla esimerkiksi tavoite, että uuden ohjelmiston myötä toimeksiantajan yrityksen liiketoiminta saastuttaa 6 kuukauden kuluttua käyttöönotosta 15 % nykyistä vähemmän. Tärkeimmät asiakkaan asettamat tavoitteet selvitetään konseptisuunnitelmaan, jonka perusteella päätetään projektin jatkosta.

Käyttjävaatimukset muodostuvat loppukäyttäjien toiveista. Loppukäyttäjiä ovat henkilöt, jotka tulevat käyttämään ohjelmistoa päivittäin. Käyttjävaatimukset ovat tavoitteita tai toimintoja, joita loppukäyttäjien pitää pystyä tekemään tuotteen avulla (Wieggers 2003, 9). Käyttjävaatimukset ovat tärkeimpiä vaatimuksia projektissa. Niiden huolellisella keräämisellä varmistetaan asiakkaan tyytyväisyys.

Käyttjävaatimuksia kootaan usein käyttötapauksien avulla. Käyttötapaus on kuvaus askelista ja vaiheista, joita käyttäjän ja ohjelman on tehtävä saavuttaakseen toivotun lopputuloksen. Käyttötapaukset ovat tehokas tapa kuvata ja havainnollistaa prosesseja sanallisesti. Käyttötapauksissa täytyy ottaa huomioon liiketoiminnan säännöt, jotka usein saattavat rajoittaa tai tarkentaa joidenkin käyttötapauksen kulkua.

Käyttötapaukset kootaan käyttötapausdokumenttiin. Käyttötapaus sisältää ainakin uniikin id-numeron, yksiselitteisen nimen, tärkeimmät tapauksen toimijat, lyhyen kuvauksen, ennakkovaatimukset (ennakkotilan), tapahtumien normaalin kulun, toivotun lopputuloksen, lopputilan, poikkeuksellisen kulun, prioriteetin ja mahdolliset virhetilanteet. Esimerkki käyttötapauksesta löytyy liitteestä numero 1.

Käyttötapausten priorisointi on tärkeää. Prioriteettien avulla voidaan määrittää tuotteen rakentamisjärjestys. Jos aikataulujen kanssa tulee ongelmia, saadaan ensimmäisiin julkaisuihin mukaan ainakin tärkeimmät ominaisuudet. Jos jotakin jää puuttumaan lopullisesta tuotteesta, on se luultavasti pienen prioriteetin ja arvon ominaisuus.

Käyttötapauksia tukemaan voidaan laatia erilaisia käyttäytymiskaavioita, kuten käyttötapaus- ja toimintokaavioita. Kaavioiden käyttö ei ole pakollista, mutta usein se helpottaa kokonaisuuden hahmottamista. Kaaviot eivät usein tarjoa lisätietoja vaatimuksista vaan kuvaavat jo tiedossa olevia vaatimuksia. Esimerkiksi käyttötapaus- ja toimintokaaviosta löytyy liitteestä numero 2.

Lisäksi toiminnallisiin vaatimuksiin kuuluu järjestelmävaatimukset. Järjestelmävaatimus on vaatimus, joka kuvaa mitä rajoitteita tai vaatimuksia järjestelmä asettaa tuotteelle. Älypuhelin sovelluksessa rajoite voisi olla esimerkiksi muistin vähyys. Edellä mainitut vaatimuksen ovat kaikki toiminnallisia vaatimuksia. Lyhyesti toiminnallinen vaatimus on toiminto, joka tuotteen täytyy tehdä ollakseen hyödyllinen käyttäjille (J. Robertson & S. Robertson 2010, 29).

2.4.2 Ei-toiminnalliset vaatimukset

Ei-toiminnallinen vaatimus on ominaisuus tai laatu, joka tuotteella on oltava (J. Robertson & S. Robertson 2010, 29). Ohjelmistomäärittelyssä voidaan mainita esimerkiksi, että uuden käyttäjän on pystyttävä ensimmäisen kerran ohjelmistoa käyttäessään, tekemään tilaus 10 minuuttia käytön aloittamisen jälkeen. Tällainen vaatimus on ei-toiminnallinen vaatimus ja tarkemmin niin kutsuttu laatuvaatimus.

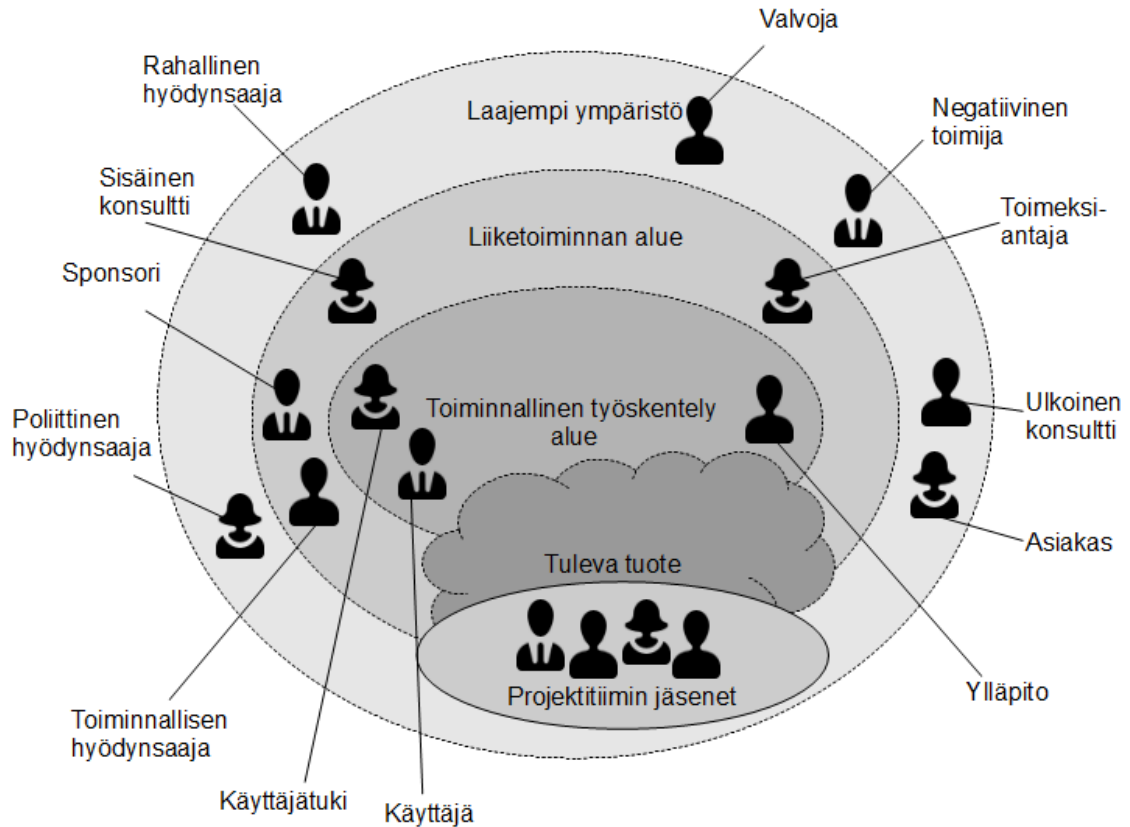
Liiketoiminnan säännöt ovat rajoitteita, jotka tulevat ohjelman ulkopuolelta. Rajoitteita voi asettaa esimerkiksi yrityksen käytänteet tai valtion asettamat säädökset. Säännöt saattavat esimerkiksi vaatia tietynlaisia raportteja tai määrittää, kuka saa tehdä tiettyjä toimintoja.

Laatuvaatimusten avulla määritetään tuotteen suorituskykyä. Laatuvaatimuksia on olemassa monia. Yleisimmin ohjelmistojen laatuvaatimukset kuitenkin koskevat käytettävyyttä, ylläpidettävyyttä, luotettavuutta, tehokkuutta ja integroitavuutta. Muita ei-toiminnallisia vaatimuksia ovat ulkoiset käyttöliittymät ja rajoitteet. Ulkoisilla käyttöliittymillä, joka useimmiten on graafinen käyttöliittymä, tarkoitetaan tuotteen yhteyttä ulkomaailmaan. Rajoitteet ohjaavat kehittäjän ratkaisuja käyttöliittymän ja suunnittelun suhteen.

Kun erilaisia määrittelyitä on riittävästi muodostetaan niistä projektin vaatimusmäärittelydokumentti. Tämän dokumentin muodostavat kaikki toiminnalliset ja ei-toiminnalliset vaatimukset. Dokumentti tulee kirjoittaa selkokielellä ja hyväksyttävä sekä asiakkaalla että tuotantotiimin muilla jäsenillä ennen tuotteen rakentamisen aloittamista. Esimerkki vaatimusmäärittelydokumentista löytyy liitteestä 5.

2.5 Määrittelyn toimijat

Jokaisessa projektissa on tietty joukko siitä kiinnostuneita tahoja. Eri toimijoiden kiinnostuksen aiheet ja vaikutusalueet ovat erilaiset. Nämä kiinnostukset ja vaikutukset on kartoitettava, jotta osataan työskennellä oikeiden ihmisten kanssa oikeaan aikaan. Isoissa projekteissa toimijoita voi olla tusinoittain. Kaikkien näiden toimijoiden löytäminen vaatii hieman työtä. Kuvion 3 toimijakartan avulla on helpompi lähteä selvittämään oman projektin toimijoita.



KUVIO 3. Toimijakartta (J. Robertson & S. Robertson 2010, 46, muokattu)

Kuten kuviosta 3 selviää toimijoita etsittäessä voidaan ajatella, että niitä on kolmella alueella projektitiimin lisäksi. On toiminnallinen alue, joka sisältää toimijoita, jotka ovat suorassa kontaktissa tuotteen kanssa, liiketoiminnan alue, joka sisältää toimijoita, jotka hyötyvät tuotteesta jollakin tavalla. Kolmantena on laajempi ympäristö, josta löytyy toimijoita, joilla on jokin vaikutus tai kiinnostus tulevaa tuotetta kohtaan. (J. Robertson & S. Robertson 2010, 46.) Käydään lyhyesti läpi kartassa esiintyviä toimijoita, keitä he ovat, mitä he tekevät ja mistä heidät tunnistaa.

Aloitetaan asiakkaasta ja toimeksiantajasta. Kun tuotetta aletaan rakentamaan, jonkun täytyy maksaa sen kehityksestä. Toimeksiantaja on henkilö, joka rahoittaa tuotteen rakennuksen, myös sponsori ajatellaan joskus toimeksiantajaksi. Asiakas on joku, joka ostaa tai käyttää tuotetta, kun se on valmis. Asiakas voi olla joukko henkilöitä, joiden nimet tiedetään tai tuhatpäinen

joukko vielä tunnistamattomia henkilöitä, jotka ostavat tuotteen kaupasta oman kiinnostuksensa pohjalta (J. Robertson & S. Robertson 2010, 48).

Käyttäjät ovat toiminnallisella alueella olevia henkilöitä, jotka ovat suorassa kosketuksessa tuotteeseen. Tuotteen on lopulta tarkoitus auttaa näitä käyttäjiä työssään, joten heidät kannattaa tuntea hyvin (J. Robertson & S. Robertson 2010, 49). Ylläpidolta voidaan selvittää tarvittavat ylläpidolliset vaatimukset tuotteelle. Käyttäjätuki osaa auttaa käyttäjävaatimusten määrittelyssä. Käyttäjätukeen kuuluvat mm. tekninen tuki, kouluttajat ja valmentajat. He ovat perillä asioista, joita tuotteiden ei kuuluisi tehdä ja asioista, joita tuotteissa on hankala oppia ja sisäistää. Kehitysehdotusten lähteinä he ovat hyviä.

Lisäksi jokaiseen projektiin liittyy erinäinen määrä toimijoita, joihin tuote vaikuttaa epäsuorasti. Tällaisia henkilöitä ovat ulkoiset ja sisäiset konsultit. He ovat jonkin alan spesialisteja, jotka tarjoavat oman tietämyksensä projektiin. Kartassa esiintyvät hyödynsaajat ovat normaalisti yrityksen johtoporrasta, joka on kiinnostunut tuotteen tuomista hyödyistä. Valvojiin taas kuuluu henkilöitä kuten turvallisuustarkastajat, palomiehet, tekniset tarkastajat tai valtion tarkastajat (J. Robertson & S. Robertson 2010, 53).

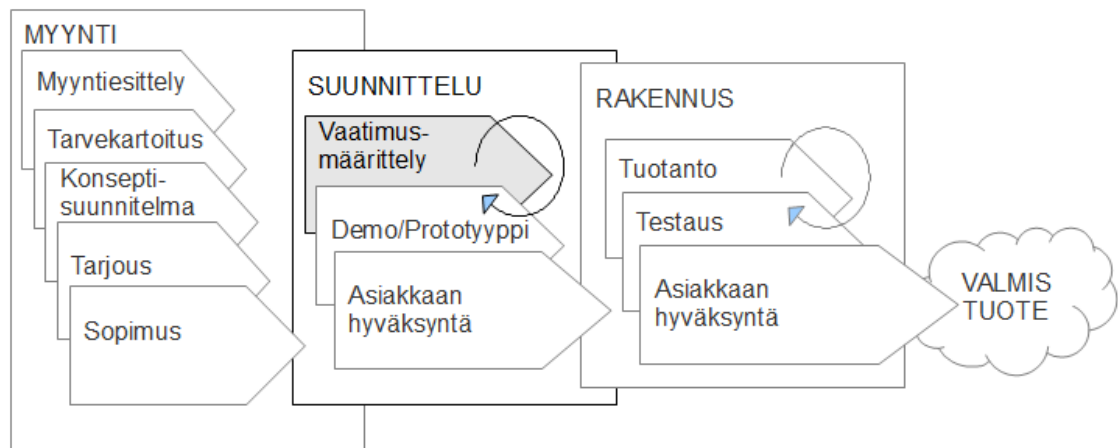
Negatiivisella toimijalla tarkoitetaan henkilöitä, jotka eivät halua tulevaa tuotetta käyttöönsä (J. Robertson & S. Robertson 2010, 53). He voivat olla tuotetta vastaan erilaisista syistä. Se saattaa esimerkiksi muuttaa heidän työnkuvaansa erilaiseksi. Tällaiset henkilöt on hyvä ottaa huomioon ja kuunnella heidän ajatuksiaan tuotteen ongelmapuolista.

Jokaiseen projektiin kuuluu tietysti myös tuotteen rakentava projektitiimi. Projektitiimin toimijoita ovat suunnittelija, ohjelmoijat, testaajat, ohjelmistoanalyttikot, ohjelmistoarkkitehdit, tekniset kirjoittajat, tietokantasuunnittelijat ja kaikki, jotka jotenkin osallistuvat tuotteen rakentamiseen (J. Robertson & S. Robertson 2010, 52). Jonkin projektitiimin jäsenen panos projektiin saattaa olla minimaalinen ja tarpeellinen vain tietyssä kohdassa kehitystä, mutta on tärkeää selvittää

etukäteen kenen osaamista tarvitaan ja missä kohtaan. Näin tuotteen aikatauluksessa ei ilmene yllätyksiä lomien tai muiden projektien aikataulujen osalta.

2.6 Määrittelyn vaiheet

Kun on selvitetty, mitä ohjelmistomäärittely oikeastaan on ja ketkä siinä ovat mukana, vilkaistaan vaatimusmäärittelyn eri vaiheita. Oheisessa kuviossa 4 on kuvattu henkilöstöhallinnan työkalujen asiakasprojektin eteneminen suunnittelusta toteutukseen työn toimeksiantajalla. Tummemmalla on suunnitteluvaihe ja siellä vaatimusmäärittely. Tässä alaluvussa puretaan vaatimusmäärittely eri vaiheisiin.

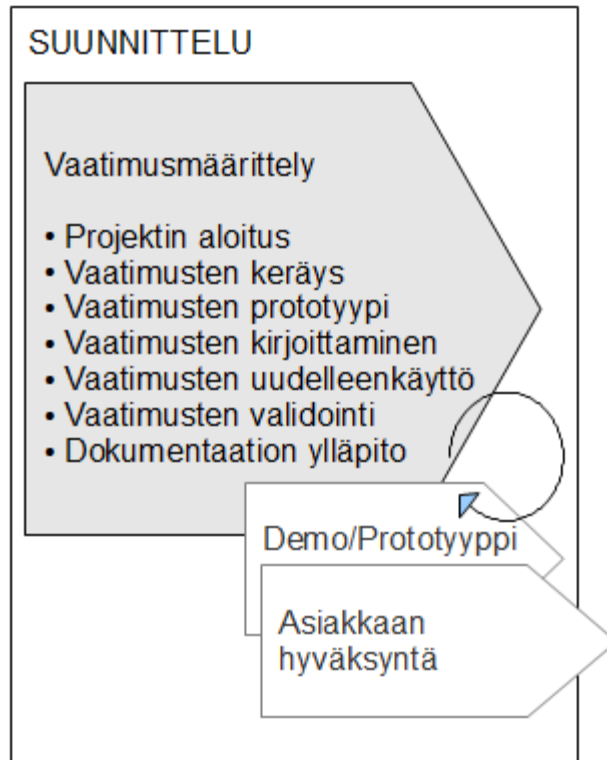


KUVIO 4. Henkilöstöhallinnan työkalujen asiakasprojektin eteneminen

Tuotteen suunnitteluvaihe aloitetaan normaalisti vasta, kun asiakkaan kanssa on tehty sopimus. Tuotteen rakennusvaihe taas aloitetaan vasta, kun tuotteen määrittely on valmis ja hyväksytetty asiakkaalla. Kuulostaa yksinkertaiselta, mutta sitä se ei aina ole.

Joskus määrittelyprosessista puhutaan, kuin se olisi prosessi, jolla on selkeä alku ja loppu: Yhtenä päivänä aloitetaan tyhjän paperin ja osaavien toimijoiden kanssa, ja kun vaatimukset tuotetta kohtaan on määritetty, työ on ohi (J. Robertson & S. Robertson 2010, 345). Todellisuudessa määrittelyprosessi

kestää koko tuotteen elinkaaren ajan, joskus tuotteen käytön loputtuakin, sen määrittelyitä kun voidaan uudelleen käyttää toisissa projekteissa. Seuraavassa kuviossa avataan tarkemmin vaatimusten määrittelyprosessia ja sen sisältämiä vaiheita.



KUVIO 5. Määrittelyprosessin vaiheet

Kuten kuviosta 5 selviää, vaatimusmäärittelyssä tärkeimmät vaiheet ovat projektin aloitus, vaatimusten keräys, vaatimusten prototyypit, vaatimusten kirjoittaminen, vaatimusten uudelleenkäyttö, vaatimusten validointi sekä dokumentaation ylläpito (J. Robertson & S. Robertson 2010, 21–29). Nämä askeleet ovat jokaisessa onnistuneessa projektissa pakollisia vaiheita. Seuraavissa alaluvuissa tarkastellaan jokaista askelta hieman lähemmin.

2.6.1 Projektin aloitus

Määrittelyprosessin alkaa esitutkimuksella, jonka tuloksista laaditaan konseptisuunnitelma. Konseptisuunnitelma määrittää pohjan projektin kehitystyölle. Siinä selvitetään asiakkaan liiketoimintavaatimukset, visiot, mahdollisuudet ja tavoitteet.

Kun perusteet projektiin ovat selvillä, on aika päättää, onko projektin toteuttaminen järkevää. Päätämisen apuna on konseptisuunnitelma. Projektin hyväksymiseksi pidetään yleensä ns. plastoff-kokous, johon osallistuu toimittajan tärkeimmät toimijat. Näihin kuuluvat yleensä projektipäällikkö, määrittelijät, testaajat ja kehittäjät. Myös yrityksen johtoa voi olla paikalla. Tässä kokouksessa arvioidaan projektin edut ja kannattavuus. Mikäli projektia päätetään jatkaa, tehdään asiakkaan kanssa sopimus ja aloitetaan tarpeiden kartoitus.

Kun projekti on aloitettu, arvioidaan projektin kustannukset ja aikataulu. Jotta näitä arvioita voidaan tehdä, on kartoitettava projektin riskit. Riskien kartoittamisen ja dokumentoinnin hyvinä puolina on se, että pystytään arvioimaan projektin menestystä paremmin ja myös se, että kaikki projektin toimijat ovat tietoisia mahdollisista riskeistä ja pystyvät tahoillaan ennaltaehkäisemään niitä.

2.6.2 Vaatimusten keräys

Jotta vaatimusten keräämisen voi aloittaa, on ymmärrettävä kuka asiakas on, mitä hän tekee työkseen ja mitä projektin on tarkoitus saavuttaa. Alkuvaiheessa on selvítettävä, mihin ongelmaan projektin lopputuloksen on tarkoitus tuoda ratkaisu. Projektin ainoana tavoitteena ei kuitenkaan tule olla ongelman ratkaiseminen, vaan myös lisähyödyn ja edun tuottaminen niin asiakkaan kuin omankin yrityksen liiketoiminnalle. Projekti ei ole kannattava, ellei alusta asti ymmärretä, mitä sen on tarkoitus saavuttaa ja mitkä ovat sen edut asiakkaan

liiketoimelle. Projekti ei ole järkevä, mikäli se ei tuota mitään etuja tai tuottaa minimaalisia hyötyjä asiakkaan liiketoiminnalle.

Vaatimusten kerääminen on aikaa vievä prosessi, joka vaatii usean kierroksen asiakkaan ja toimittajan välillä. Vaatimusten määrittelyyn on otettava mukaan asiakkaan tuotteen kannalta tärkeimmät toimijat. Heidän kanssaan on hyvä tavata vähintään kerran kasvotusten ja keskustella tulevasta tuotteesta. Tällaisissa palavereissa taitavan määrittelijän on helppo ohjata keskustelua haluamaansa suuntaan ja näin löytää tarvittavia vaatimuksia.

Yleisimpiä vaatimusten keräämisen välineitä ovat haastattelut, prototyypit ja demoamispalaverit. Haastatteluiden avulla selvitetään, kuka asiakas on ja mitä hän tekee työkseen. Määrittelyn kannalta asiakkaan työn ymmärtäminen on tärkeää. Jos ei ymmärrä, mitä asiakas tekee työkseen, on hankalaa suunnitella tuote, josta olisi asiakkaalle hyötyä. Lisäksi yksi määrittelijän tärkeimmistä taidoista ja työkaluista on kuunteleminen. On kuunneltava, mitä asiakas kertoo.

Kun asiakkaille näyttää jotain konkreettista, alkaa vaatimuksiakin yleensä löytyä. Haastatteluihin ja palavereihin täytyy mennä valmistautuneena ja niissä on hyvä välttää tekemästä ehdotuksia. Asiakkailta kun on yleensä tapana olla samaa mieltä toimittajan tekemistä ehdotuksista. Keskustelua on yritettävä johdattaa niin, että asiakas saa itse pohtia ja löytää haluamansa ratkaisun kuhunkin ongelmaan.

Vaatimuksia kartoittaessa lisäksi hyödyllisiä keinoja ovat asiakkaan työskentelyn seuraaminen oikeiden töiden parissa tai työskentely itse asiakkaalle ominaisten töiden parissa. Kun ymmärtää, mitä asiakas tekee, on myös helpompi ymmärtää mitä asiakas tarvitsee. Näin vaatimuksista tulee todellisuudessa arvokkaita myös asiakkaalle, eikä aleta tekemään ominaisuuksia, jotka ovat hienoja, mutta hyödyllisiä vain toimittajan mielestä.

2.6.3 Vaatimusten prototyypit

Prototyypit eivät ole hyviä ainoastaan vaatimuksia etsittäessä, vaan myös niitä testattaessa. Kun vaatimuksia on löydetty tarpeellinen määrä, voidaan niitä testata tekemällä niiden pohjalta prototyyppejä. Prototyypit voivat olla yksinkertaisimmillaan piirustuksia jostakin halutusta ominaisuudesta, tai joitakin toiminnallisuuksia sisältäviä ohjelmia.

Prototyypit ovat erityisen hyviä silloin kun kumpikaan, asiakas tai toimittaja, eivät ole varmoja, mitä lopullisen tuotteen tulisi sisältää. Prototyyppejä käytettäessä on kuitenkin tärkeää muistaa ja selventää myös asiakkaalle, että kyseessä on vain prototyyppi, ei lopullinen tuote. Joskus prototyypit ovat niin hyviä että asiakas haluaa juuri prototyypin näköisen tuotteen. Prototyypit olisikin hyvä pitää toiminnallisuuksissa ja välttää graafisen ulkoasun tekoa, joka antaisi tietyn kuvan siitä, miltä lopullinen tuote tulee näyttämään.

Asiakkaan täytyy myös ymmärtää, että jokin toiminnallisuus ei tule välttämättä vastaamaan prototyypin esittämää toiminnallisuutta askel askeleelta. Se ei ole prototyypin tarkoitus. Prototyypeillä yritetään varmentaa kehityksen suunta ennen varsinaisen ohjelmointityön aloitusta.

Prototyypin rakentamiseen ei kannata käyttää paljoa aikaa ja vaivaa. Prototyyppi on aluksi vain ehdotus, jolla esitellään joitakin ideoita ja mahdollisia ratkaisuja. Prototyyppi muuttuu jatkuvasti kehityksen myötä ja ottaa uusia muotoja. Prototyyppien laatiminen on iteratiivista toimintaa, jossa samoja toimintoja tehdään uudelleen ja uudelleen kunnes saavutetaan haluttu tai tyydyttävä lopputulos.

2.6.4 Vaatimusten kirjoittaminen

Pelkkä vaatimusten selvittäminen ei riitä, ne täytyy myös dokumentoida. Vaatimuksia säilötään yleisesti erilaisiin dokumentteihin ja kaavioihin, mutta tietoa voi olla myös esimerkiksi kuvina ja tietokannoissa. Määrittelyprosessin aikana dokumentit kehittyvät hitaasti yhtenäiseksi kokonaisuudeksi.

Vaatimusten kirjoittaminen tehdään pala ja vaatimus kerrallaan projektin edessä. Jokainen vaatimus ei välttämättä ole hioutunut tarkaksi määritteeksi, mutta vaatimuksia kirjoittaessa se täytyy muuntaa ymmärrettävään muotoon. Kirjoitettujen vaatimusten tarkoituksena on rakentaa yhteinen ymmärrys kaikkien toimijoiden välille siitä, mitä lähdetään rakentamaan.

Lopulta syntyvät dokumentit kertovat, mitä aletaan rakentaa. Kaikkien toimijoiden on hyväksyttävä vaatimukset ja oltava yhteisymmärryksessä niistä. Vaatimuksia dokumentoidessa on tärkeää käyttää selkokieltä ja muotoilla asiat mahdollisemman lyhyesti ja selkeästi. Alakohtaista jargonia tulee välttää. Selkokieli-syydellä varmistetaan, että kaikki toimijat, myös he joilta tekninen ymmärrys puuttuu, ymmärtävät mitä dokumenteissa sanotaan.

Projektin eri dokumentit kertovat eri asioita projektin vaatimuksista. Konsepti-suunnitelmassa on selvitetty projektin liiketoiminnalliset vaatimukset, eli mitä hyötyä asiakas hakee tuotteelta. Käyttjävaatimukset kirjataan yleensä käyttötapausdokumenttiin ja toiminnalliset sekä ei-toiminnalliset vaatimukset vaatimusmäärittelydokumenttiin, joka lopulta kokoaa kaiken yhteen. Lisäksi vaatimuksia voi säilöä kaavioihin, kuviin ja tietokantoihin. Kaiken ei tarvitse olla kirjoitetussa muodossa.

Kun vaatimukset on dokumentoitu ja niistä ollaan yhtämieltä, tehdään dokumenteista ns. baseline-dokumentteja. Baseline-dokumentit tai dokumentti kertovat, mitä on sovittu rakennettavaksi. Kun tuotteen rakennus on alkanut, sen jälkeen

ilmeneviä pyyntöjä ja vaatimuksia verrataan baseline-dokumenteista löytyviin vaatimuksiin. Näiden dokumenttien perusteella arvioidaan jokaisen uuden vaatimuksen vaikutus projektiin ja päätetään aletaanko sitä toteuttaa. Määrittely on siis tärkeää niin tuotteen rakentamisen kuin sen hallittavuudenkin kannalta. Laadukas määrittely maksaa itsensä takaisin moneen kertaan - tuotteen rakentaminen on tarkempaa, ylläpidon kustannukset ovat alhaisemmat ja lopullinen tuote vastaa tarkasti asiakkaan toiveita ja tarpeita (J. Robertson & S. Robertson 2010, 257).

2.6.5 Vaatimusten uudelleenkäyttö

Ei ole järkeä kirjoittaa uudelleen vaatimuksia, jos joku on ne jo aiemmassa projektissa dokumentoinut. On siis järkevää käydä läpi vastaavien projektien dokumentaatiota ja tutkia löytyisikö sieltä jotakin käyttökelpoista. Vaatimusten uudelleen käyttö säästää aikaa ja resursseja.

Vaatimuksia ei ole mahdollista löytää kaikkia yhdeltä istumalta. Ei tule myöskään koskaan olettaa, että vaatimusmäärittely on täysin valmis. Jokainen projekti elää elinkaarensa aikana ja voi ottaa uusia suuntia koska vain. Kun vaatimuksia kuitenkin on projektin etenemisen kannalta tarpeeksi ja nähdään että voidaan edetä, on seuraava vaihe vaatimusten validointi.

2.6.6 Vaatimusten validointi

Vaatimuksia validoidaan, jotta kehittäjät eivät joutuisi työskentelemään monimerkityksisten ja keskeneräisten vaatimusten kanssa. Jos määrittelyt ovat vaja- vaiiset tai sekavat, joutuu kehittäjä itse kehittämään ratkaisun tai tulkinnan ongelmaan ja ratkaisu ei yleensä ole sitä, mitä asiakas on alun perin pyytänyt.

Validoinnilla pyritään karsimaan myös turhat, vähän arvoa tuovat vaatimukset pois. Ian Alexanderin laatiman listan avulla on helppo tarkistaa oman projektinsa vaatimukset. Jokaisen vaatimuksen kohdalla kysytään onko se selkeä, onko se niin lyhyt kuin mahdollista, päteekö se tiettyyn käyttäjään, onko sillä järkevä prioriteetti, onko se todennettavissa, onko se yksi vaatimus, onko sen alkuperä selvillä, onko sillä ainutlaatuinen tunniste (Alexander 2002, 78–79).

2.6.7 Dokumentaation ylläpito

Kun vaatimusmäärittely on saatu kuntoon ja tuotteen rakentaminen aloitettu, muuttuu määrittelyn kanssa työskentely monen mielestä epämiellyttäväksi ja jopa turhaksi. Dokumentaation ylläpito on kuitenkin välttämätön paha jokaisessa projektissa. Jollei dokumentteja ylläpidetä, muutoksista ja uudistuksista pidetä kirjaa, ollaan pian hukassa projektin alkuperäisen tavoitteen ja vaatimusten kanssa.

Dokumentoimattomat muutokset johtavat nopeasti syrjäytyneisiin vaatimuksiin ja epäselvyyteen projektin kehityksen suunnasta. Jokainen projekti reagoi hallitsemattomiin muutoksiin omalla tavallaan. Useimmiten vaikutukset näkyvät aikataulun pettämisenä, budjetin ylittymisenä tai laadun laskemisena.

Kun teoriassa on selvillä miten onnistuneen vaatimusmäärittely prosessin pitäisi edetä, voidaan siirtyä tarkastelemaan työn toimeksiantajan vaatimusmäärittelyn nykytilaa.

3 NYKYTILA

Määrittelyprosessi työn toimeksiantajalla on nykyisellään kiitettävällä tasolla. Ongelmaa ei nähdäkään siinä, ettei vaatimuksia löydettäisi, vaan siinä, että niitä ei dokumentoida ja ylläpidetä. Niin sanotussa normaalissa tilassa määrittelydokumentaation nykytila ei aiheuta suuria ongelmia. Normaalina tilana voidaan pitää tilaa, jossa projektitiimin jäsenet ovat vakiintuneet ja työskennelleet asiakkaiden ja projektien parissa pidemmän aikaa. Silloin vaatimukset pysyvät hallinnassa ja projektit etenevät. Kun normaalitila muuttuu henkilöstövaihdosten, tai muiden yllättävien tapahtumien myötä, aiheuttaa se ongelmia projektien etenemisessä.

3.1 Määrittelyn nykytila toimeksiantajalla

Nykyisellään ohjelmistojen määrittelydokumentaatiot ovat epämääräisiä ja niiden määrä, sekä laatu vaihtelevaa. Projektien välillä dokumenttien laatu ja tarkkuus voi vaihdella suuresti. Osaltaan tähän vaikuttaa asiakasyrityksen koko, tiimin sen hetkinen koostumus, sekä määrittelyn hetkellä muiden projektien määrä ja aikataulutus.

Kun asiakas on pienempi yritys ja heidän projektiryhmänsä koostuu joskus jopa yhdestä henkilöstä, jää määrittelydokumenttien teko usein vähemmälle, koska ei asiakas niitä vaadi. Jos taas asiakkaana on iso yritys ison projektiryhmän ja tarkkojen laatuvaatimusten sekä rajoitteiden kanssa, vaaditaan usein kaikki määrittelyt yksityiskohtaisesti paperille. Määrittelyiden tekohetkellä myös tiimin jäsenet vaikuttavat lopputulokseen. Jos esimerkiksi henkilöstössä on vajausta, ei ehditä panostamaan jokaiseen projektiin täysillä. Myös kiireinen aikataulu vie aikaa pois projektien määrittelyiden dokumentoinnilta.

Dokumenttien laatua suurempi ongelma on kokonaan puuttuvat dokumentit. Monessa projektissa on luotettu siihen, että ihmiset, jotka ovat osa projektiryhmää tietävät, mitä ovat tekemässä ja miksi mitään tehdään. Ongelmat alkavat, kun normaalitilasta poiketaan ja projekti jatkuu uusien projektiryhmäläisten kanssa. Koska määrittelyt ovat edellisten työntekijöiden päässä, kahvitauon keskusteluissa ja sähköposteissa, joudutaan asiakkaalta uudelleen kerta toisensa jälkeen kyselemään, mitä he tuotteelta tarvitsevat.

Jokaisen projektin kohdalla tulisi olettaa, että jokin menee vikaan. Henkilöstö vaihtuu, ihmiset lomailevat, sairastuvat ja tekevät virheitä. Nykyisellään ongelmien ilmetessä ei voida taata projektien sulavaa etenemistä, koska vaatimuksista ei ole tarpeellista dokumentointia olemassa. Ne dokumentit, jotka ovat olemassa, löytyvät usein sähköpostien syövereistä tai yhden työntekijän kotikansiosta. Ei ole yleistä tallennuspaikkaa ja rakennetta, johon kaikilla projektiryhmäläisillä olisi halutessaan pääsy. Ongelmana on siis myös dokumenttien saatavuus ja vähyys.

3.2 SWOT- analyysi

SWOT-analyysi on Albert Humpreyn kehittämä menetelmä (Gül 2008, 46), jolla tunnistetaan ongelmia ja mahdollisuuksia yritysmaailmassa. Lyhenne SWOT tulee sanoista vahvuudet (strengths), heikkoudet (weaknesses), mahdollisuudet (opportunities) ja uhat (threats). Usein analyysiä käytetään kuvaamaan yrityksen tilaa markkinoilla, mutta sitä voidaan myös soveltaa projektien ja hankkeiden suunnittelussa. Alla on kuvattu toimeksiantajani ohjelmistomäärittelyn tila SWOT-analyysin avulla. Analyysin avulla pyritään selittämään ongelmakohdat, jotta niihin voidaan tarttua.

TAULUKKO 1. SWOT-analyysi työn toimeksiantajan ohjelmistomäärittelystä

SWOT- analyysi ohjelmistomäärittelystä	
Vahvuudet	Heikkoudet
<ul style="list-style-type: none"> • Kokenut tiimi • Toimivat asiakassuhteet • Toimivat perustuotteet • Toimiva määrittelyprosessi 	<ul style="list-style-type: none"> • Dokumentoinnin puute • Dokumentoinnin ylläpidon puute • Dokumentoinnin epäjohtamukaisuus • Dokumenttien saatavuus
Mahdollisuudet	Uhat
<ul style="list-style-type: none"> • Parantaa imagoa/mainetta • Sitouttaa henkilökuntaa projekteihin • Aikataulussa pysyminen • Budjetissa pysyminen 	<ul style="list-style-type: none"> • Dokumentoinnin tilanne heikentyy • Imago/maine heikentyy • Markkinaosuuden menetys

Seuraavissa alaluvuissa SWOT-analyysin osiot ja luettelot puretaan omiksi luvuikseen tarkastelua varten. Heikkouksia analysoidaan tarkemmin kuin muita osioita, koska niiden avulla pystytään tunnistamaan ongelmakohtia ja kehitysideoita.

3.2.1 Vahvuudet

Vahvuuksiin kuuluu, että yrityksessä on kokenut tiimi, jolla on paljon kokemusta alan tarpeista ja vaatimuksista. Kokemuksen avulla vältetään suurimmilta ongelmilta, joita projektit kohtaavat määrittelyn puuttuessa tai sen ollessa vajavainen. Toimivat asiakassuhteet auttavat ongelmatilanteissa, koska vuorovaikutus asiakkaiden kanssa on toimivaa ja pahoilta erimiellisyksiltä vältetään.

Toimivien perustuotteiden avulla voidaan taata tuotteille tietty perustaso, joka on riippumaton asiakaskohtaisista määrittelyistä. Näin vältetään projektien peruuntumiselta ja asiakkaan suurelta tyytymättömyydeltä. Toimiva määrittelyprosessi takaa projektien etenemisen, jos yritys on normaalissa tilassa (ks. 3

Nykytila). Vaatimukset löydetään tehokkaasti ja ne saadaan toteutukseen nopeasti.

3.2.2 Heikkoudet

Heikkouksia ovat dokumentoinnin ja ylläpidon puute, dokumentoinnin epäjohdonmukaisuus sekä dokumentoinnin saatavuus. Dokumentoinnin puute aiheuttaa sen että rakennus- ja testausvaiheessa saatetaan joutua tekemään useita korjauskierroksia asiakkaiden palautteiden avulla. Aina löytyy jotakin, mikä ei asiakasta miellytä tai on toteutettu väärin, koska dokumentointi on vajaata.

Dokumentoinnin ylläpidon puute aiheuttaa sen, että muutoksien kertyessä kukaan ei enää tiedä, mitä on alunperin luvattu asiakkaalle, miksi tietty toiminnallisuus on tehty ja mitkä ominaisuudet ovat uusia ja kehitystyön piiriin kuuluvia laskutettavia toiminnallisuuksia. Kun dokumentit eivät ole ajan tasalla, myös ylläpidon työ hankaloituu. Suurimmat ongelmat esiintyvät laskutuksessa ja ylläpidossa.

Dokumentoinnin epäjohdonmukaisuuden takia jokaisesta projektista löytyy sekalainen kirjo dokumentteja, joita ovat laatineet eri henkilöt projektin aikana. Kahdesta eri projektista sama tieto saattaa löytyä aivan erilaisesta dokumentista aivan eri paikasta. Joku saattaa kirjoittaa pitkän dokumentin, kun joku hoitaa määrittelyt kaavioilla ja havainnoillistavilla kuvilla. Kun dokumentaatiot eivät ole yhtenäisiä, hidastuu etsittävän tiedon etsiminen huomattavasti.

Heikkouksissa dokumenttien saatavuudella tarkoitetaan yleisen tallennusrakenteen ja arkistointikäytänteiden puuttumista. Paljon dokumentteja on yhteisessä verkkokansiossa, mutta niin että vain tietyillä henkilöillä on niihin pääsy. Valitettavan usein dokumentit jäävät myös pyörimään ainoastaan

sähköpostiin, koska ei ole selvää, mihin ne olisi syytä tallentaa. Tämä aiheuttaa hämmennystä siitä, onko jokin dokumentti olemassa, jos niin missä ja keneltä sitä pitää pyytää. Vaikka määrittelyn dokumentoinnissa onkin ongelmia, ongelmat eivät kaada kehitystä vaan hidastavat projektien parissa työskentelyä ja aiheuttavat ylimääräisiä kustannuksia.

3.2.3 Mahdollisuudet

Mahdollisuuksiin lukeutuvat yrityksen imagon/maineen parantaminen ja henkilöstön sitouttaminen projekteihin. Lisäksi onnistuneen kehitystyön avulla voidaan saavuttaa säästöjä niin ajallisesti kuin rahallisesti. Nämä mahdollisuudet olisi saavutettavissa kehittämällä määrittelyprosessia ja dokumentaatiota. Kehitys nostaa projektien onnistumisprosenttia ja asiakkaiden tyytyväisyyttä, tämä puolestaan parantaa yrityksen imagoa ja mainetta markkinoilla. Kun projekteissa menestytään paremmin ja toiminta on hallittua, on henkilöstöllä korkeampi motiivi tehdä töitä projektien ja onnistumisen eteen.

3.2.4 Uhat

Uhkina voidaan nähdä imagon ja maineen heikkeneminen, jolloin asiakkaat menettävät luottamuksen yritykseen. Jos asiakkaat menettävät luottamuksensa yritykseen, etsivät he jatkossa tarvitsemilleen palveluille toisen tarjoajan. Huono dokumentointi voi aiheuttaa esimerkiksi tilanteita, joissa heikko dokumentaatio aiheuttaa toiminnallisuuksien puuttumisia lopullisesta tuotteesta, vaikka ne on luvattu toteuttaa. Tällainen ei ymmärrettävästi herätä asiakkaiden luottamusta yrityksen toimintaan. Huono määrittely aiheuttaa myös rahallisia kustannuksia aikataulujen ylityksinä ja uudelleen työstämisenä.

3.3 Ongelmakohtat

Analyysin perusteella ongelmakohtina voidaan pitää asiakkaiden tarpeiden dokumentointia ja dokumentaation ylläpitoa. Kun projekteista ei laadita kunnollisia dokumentteja, projektit eivät valmistu ajallaan, matalalla virheprosentilla ja toivotulla nopealla tahdilla. Lisäksi tuotteiden ylläpidettävyys laskee, kun määrittelyt eivät ole ajan tasalla.

Määrittelyprosessissa itsessään ei ole suuria ongelmia. Projektien ongelmat alkavat yleensä silloin, kun löydettyjä vaatimuksia aletaan dokumentoimaan. Jokaisella on oma tapansa kirjoittaa ja havainnoillistaa ymmärtämäänsä, omia ja asiakkaan ajatuksia. Jokainen laatii eri määrän erilaisia dokumentteja tekstien, kuvien ja kaavioiden muodossa. Jokainen myös ylläpitää niitä omalla tapaansa. Tämä on yksi ongelmakohta, joka kaipaa yhdenmukaistamista. Tähän voidaan tarttua yhteisien dokumenttipohjien sekä dokumentoinnin ohjeistuksen laatimisella. Nykyisellään ei ole olemassa mitään valmiita pohjia tai ohjeistusta kuinka tulisi toimia määrittelyn eri vaiheissa.

Otetaan kuviteltu tilanne, jossa projektia X on määrittelemässä kaksi henkilöä, rouva A ja herra B. Rouva A dokumentoi projektin X alkaessa tarvittavat roolit ja oikeudet ja tallentaa ne omaan kotikansioonsa. Herra B kerää sähköpostin välityksellä asiakkaalta projektin vaatimat käyttötapaukset ja kertoo ne projektiryhmälle. Projekti etenee hyvin, kun rouva A ja herra B ohjaavat tuotantotiimiä yhteistuumin oikeaan suuntaan. Asiakaskin on tyytyväinen projektin etenemiseen.

Sitten alkaa tapahtua. Rouva A jää pitkälle sairauslomalle ja rouva C palkataan yritykseen. Herra B:llä alkaa olla vaikeuksia vastalla tuotantotiimin kysymyksiin tarpeiden osalta, jotka rouva A aikoinaan kartoitti. Samalla hän yrittää perehdyttää rouva C:tä projektin X saloihin. Kun herra B työllistyy toiseen yritykseen, jää rouva C ja projektiryhmä arvailemaan projektin X vaatimuksia. Lopulta rouva C joutuu selvittämään vaatimukset uudelleen asiakkaalta. Projekti

valmistuu reilusti myöhässä, budjetti ylitettynä ja vähimmillä toiminnallisuuksilla, mitä asiakas alunperin halusi.

Vaikka kaikki sujuisikin aluksi hyvin ilman dokumentaatiota saman tiimin ollessa projektissa mukana alusta asti, täytyy muistaa että projektiryhmään saattaa tulla muutoksia koska tahansa. Ohjelmistokehittäjät ovat kuitenkin vain ihmisiä.

Tältä kaikelta olisi voitu välttyä, jos alusta alkaen projektin X määrittelyt olisi dokumentoitu järjestelmällisesti ja niitä olisi ylläpidetty tunnollisesti. Tarinan kaltaisiin ongelmiin törmätään myös toimeksiantajallani. Liian usein joudutaan uudelleen varmistamaan asiakkaalta, mitä he alunperin halusivatkaan ja onko näistä haluista päästy yhteisymmärrykseen.

4 TAVOITETILA

Tässä luvussa kartoitetaan työn toimeksiantajan määrittelyprosessin ja -dokumentaation tavoitetila. Edellisessä luvussa tunnistettujen ongelmakohtien avulla tunnistetaan kehityskohteet. Lisäksi teoriaosuudessa kuvatun prosessin ja nykytilan vertailu tarjoavat kehityskohteita. Yhdessä kaikki löydetyt kehityskohteet ja tavoitteet muodostavat tavoitetilan määrittelyprosessille ja -dokumentaatiolle.

4.1 Kehityskohteet

Kehityskohteina ovat edellisessä luvussa tunnistetut ongelmakohtat ja niiden korjaamiseksi vaadittavat toimet. Tunnistettuja ongelmakohtia ovat dokumentoinnin ja dokumenttien ylläpidon puute, sekä dokumentaation saatavuusongelmat. Tavoitteena on tarttua tunnistettuihin ongelmakohtiin ja erilaisten toimien avulla korjata sekä ehkäistä niiden aiheuttamia ongelmia. Opinnäytetyön tarkoituksena on kehittää määrittelyprosessia dokumenttipohjien, käytänteiden ja määrittelyiden saatavuuden osalta.

Määrittelydokumentaation ja sen ylläpidon puutteisiin pyritään vaikuttamaan yhteisillä dokumenttipohjilla sekä dokumentoinnin tekemiseen ja ylläpitoon liittyvällä ohjeistuksella. Laadittavia dokumenttipohjia ovat konseptisuunnitelma, käyttötapaukset, sekä vaatimusmäärittely. Nämä pohjat laaditaan kattaviksi ja tarkoitus on, että niistä muokataan projektikohtaisesti sopivia tarkkuudeltaan ja sisällöltään.

Dokumenttien ylläpitoa pyritään parantamaan ohjeistuksella, joka ottaa kantaa muutosten ja uusien vaatimusten käsittelyyn. Ohjeistuksen avulla ei ole tarkoitus pystyä oppimaan kaikkea määrittelyprosessiin liittyvää, vaan se toimii

osana opinnäytetyötä ja yhdessä sen kanssa tarjoaa ohjeet toimeksiantajan toimintamalliin.

Määrittelyn saatavuuteen pyritään vaikuttamaan tarkistamalla roolitukset ja verkkokansioiden oikeudet. Ohjeistuksen avulla pyritään myös varmistamaan dokumenttien oikeat tallennuspaikat. Ohjeistuksella pyritään varmistamaan, että tarvittavat dokumentit löytyvät myös sähköisessä muodossa Lerlin tracista, jotta kaikki halukkaat, joilla on tunnukset traciin, saavat käsiinsä projektien viimeisimmät versiot määrittelyistä.

Opinnäytetyön tekemisen aikaan markkinoinninosasto, yhdessä yrityksen johdon kanssa, alkoi selvittää työn toimeksiantajan yhteisen verkkokansion eli Y-kansion oikeuksia ja rakennetta. Roolituksissa ja uusissa rakenteissa otetaan huomioon tuotantotiimin pyynnöt jakaa oikeuksia laajemmin asiakaskohtaisiin tietoihin. Henkilöstöhallinnan työkalujen tiimille luotiin oma hakemisto, johon sijoitettiin myös luodut asiakirjapohjat. Määrittelyiden saatavuuden katsotaan olevan riittävä, kun ne löytyvät Lerlin tracin wikistä. Tarkemmat ohjeet löytyvät määrittelydokumentaation ohjeesta, liitteestä numero 3.

4.2 Vaatimukset

Ohessa luettelo vaatimuksista, jotka pyritään saavuttamaan tavoitetilaan pääsemiseksi. Vaatimuksien saavuttamiseksi tehtävät toimet löytyvät luvusta 5.

- Ohjelmistoa määriteltäessä projektiryhmällä oltava selkeä käsitys vastuualueista.
- Ohjelmistomäärittelyt saatava tilaan, jossa projektista etsittävän tiedon on löydettävä etsijästä riippumatta.

- Ohjelmistomäärittelyt saatava tilaan, jossa projektin eteneminen ja jatkuminen ei vaarannu, mikäli projektiryhmässä tapahtuu muutoksia.
- Ohjelmistomäärittelyt saatava tilaan, jossa asiakkaalta ei tarvitse varmistaa uudelleen jo kerran selvitettyjä vaatimuksia.
- Ohjelmistomäärittelyt saatava tilaan, jossa kaikki toimijat ovat projektin alussa yhteisymmärryksessä rakennettavasta tuotteesta ja sen ominaisuuksista.
- Ohjelmistomäärittelyt saatava tilaan, jossa projektin valmistumiselle on selkeät kriteerit.
- Toteutettavat vaatimukset ja toiminnallisuudet on oltava jäljitettävissä.
- Muutosten ja uusien toiminnallisuuksien käsittely on oltava hallittua.

4.3 Ongelmakohtat ja rajoitteet

Kehitystyö rajoitetaan käsittämään henkilöstöhallinnan työkalujen vaatimusmäärittelyn dokumentaation ja niihin liittyvän arkistoinnin ohjeistukseen. Jo olemassa olevat määrittelydokumentit rajataan kehittämisprosessin ulkopuolelle. Yhteisen verkkokansion eli Y-kansion roolitukset ja rakenteen muutokset olivat opinnäytetyön teon aikaan työn alla, joten Y-kansion oikeuksiin ei oteta tarkemmin kantaa.

5 KEHITTÄMISPROSESSI

Edellisessä luvussa tunnistettiin kehityskohteita ja laadittiin luettelo vaatimuksista, jotka tulisi saavuttaa. Seuraavissa alaluvuissa katsotaan ensin tehtäviä kehitystoimenpiteitä ja niiden soveltamista vaatimusten saavuttamiseksi.

5.1 Toteutettavat asiakirjapohjat

Jotta asiakasprojektien vaatimusmäärittelyistä saataisiin yhtenäisimpiä, toteutetaan pieni joukko määrittelyssä auttavia asiakirjapohjia. Opinnäytetyön pohjalta on päätetty toteuttaa seuraavat asiakirjapohjat: konseptisuunnitelma (liite 4), käyttötapaukset (liite 1) ja vaatimusmäärittely (liite 5).

Asiakirjapohjat laaditaan kattaviksi ja niitä tulisi käyttää soveltaen eri projekteissa. Asiakirjapohjat sisältävät lyhyet kuvaukset jokaisen kohdan tarkoitettusta sisällöstä. Projektikohtaisesti käytetään UML kaavioita, kuvia ja prototyyppejä tukemaan vaatimusmäärittelyitä. Esimerkkejä kaavioista löytyy liitteestä numero 2.

5.2 Toteutettava ohjeistus

Asiakasprojektien vaatimusmäärittelyn dokumentaatiosta laaditaan ohjeistus (liite 3), josta selviävät tarpeelliset dokumentit, vaatimusten saatavuuden varmistaminen, tiimin vastuut, muutostenhallinta ja tikettikäytännöt. Ohjeistuksen avulla pyritään varmistamaan dokumenttien yhdenmukaisuus, ajantasaisuus ja saatavuus.

Ohjeistuksen ei ole tarkoitus pakottaa henkilöstöä tiettyyn kaavaan, vaan tarjota ohjeistus hyviin toimintamalleihin, joita on suositeltavaa noudattaa mahdollisuuksien mukaan. Jossain projekteissa ohjeita voi olla lähes mahdoton noudattaa tarkasti ja joihinkin ne saattavat soveltua erinomaisesti.

5.3 Tavoitetilan saavuttaminen

Luvussa neljä (4) esiteltiin luettelo vaatimuksista tavoitetilan saavuttamiseksi (4.2 Vaatimukset). Tässä alaluvussa kuvataan tehtävät toimet, luettelon kohta kerrallaan, tavoitetilan saavuttamiseksi. Osassa vaatimuksista sovelletaan samoja ratkaisua kuin muissa kohdissa.

Ohjelmistoa määriteltäessä projektiryhmällä on oltava selkeä käsitys vastuualueista. Jos ei käsitystä ole, saatetaan tehdä turhaa työtä, tai jokin tärkeä työ voi jäädä kokonaan tekemättä. Tämä pyritään välttämään yleisellä ohjeistuksella, joka sisältää myös RACI-taulukon. RACI taulukolla kuvataan tiettyä tehtävää tai työtä kohden olevat vastuulliset henkilöt. Tarkempaa tietoa RACI-taulukosta löytyy ohjeistuksesta, liitteestä numero 3.

Ohjelmistomäärittelyt on saatava tilaan, jossa projektista etsittävän tiedon on löydyttävä etsijästä riippumatta. Mahdollisia tiedon hakijoita voivat olla määrittelijät, ohjelmoijat, testaajat, projektipäälliköt, konsernipalvelut ja tekninen tuki. Kaikilla on oltava pääsy tarvittaviin tietoihin, tai tieto siitä kenen takana tiedot ovat. Tietojen saatavuutta varmistetaan ohjeistuksella, josta selviää mihin mitään tietoja pitää tallentaa ja dokumenttien vastuuhenkilöt.

Ohjelmistomäärittelyt on saatava tilaan, jossa projektin eteneminen ja jatkuminen ei vaarannu, mikäli projektiryhmässä tapahtuu muutoksia. Määrittelydokumentaatiot on oltava ajan tasalla ja saatavilla. Dokumenttipohjat

ja niiden ohjeet on oltava projektiryhmän saatavilla. Näin pyritään varmistamaan dokumenttien ajantasaisuus ja laatu.

Ohjelmistomäärittelyt saatava tilaan, jossa asiakkaalta ei tarvitse varmistaa uudelleen jo kerran selvitettyjä vaatimuksia. Kyseinen tarve liittyy samoihin asioihin kuin edelliset kohdat, eli dokumenttien saatavuuteen ja ajantasaisuuteen. Ohjeistuksella pyritään varmistamaan, että tiedot ovat oikeassa paikassa ja ajan tasalla.

Ohjelmistomäärittelyt saatava tilaan, jossa kaikki toimijat ovat projektin alussa yhteisymmärryksessä rakennettavasta tuotteesta ja sen ominaisuuksista. Tavoite pyritään saavuttamaan varmistamalla, että dokumentit on laadittu selkokielisiksi. Ohjeita määrittelyiden laatimiseen löytyy opinnäytetyön luvusta kaksi (2), sekä yleisestä ohjeesta.

Ohjelmistomäärittelyt on saatava tilaan, jossa projektin valmistumiselle on selkeät kriteerit. Selkeät valmistuskriteerit muodostuvat osaksi myös siitä, että tiedetään mitä aletaan rakentaa, mutta myös siitä, että projektin alkumäärittelyt lyödään lukkoon. Lukkoon lyömisellä tarkoitetaan dokumenttien ”baselinaamista”, kun kaikki ovat samaa mieltä vaatimuksista ja projekti voidaan aloittaa. Tarkempaa tietoa baseline-dokumenteista löytyy alaluvusta 2.6.4 sekä yleisestä ohjeistuksesta.

Toteutettavat vaatimukset ja toiminnallisuudet on oltava jäljitettävissä. Yleisestä ohjeistuksesta löytyy kohta toiminnallisuuksien purkamisesta tiketeiksi ja muista työn toimeksiantajan tikettikäytänteistä.

Muutosten ja uusien toiminnallisuuksien käsittely on oltava hallittua. Yleisestä ohjeistuksesta löytyy ohjeet muutosten ja uusien ominaisuuksien hallittuun käsittelyyn.

Lähes kaikkiin tavoitetilan vaatimukseen pyritään pääsemään asiakirjapohjien ja ohjeistuksen avulla. Jotta myös itse määrittelyprosessia saataisiin kehitettyä, suositellaan työn toimeksiantajan tuotantotiimin jäsenille opinnäytetyön teoriaosuuden, eli luvun kaksi (2) lukemista.

6 KEHITTÄMISTOIMINNAN TULOKSET

Opinnäytetyön alkaessa keskeisimmät tutkimuskysymykset olivat: Miksi määrittelyprosessi ja -dokumentaatio eivät ole toimivia tällä hetkellä? Mitkä toimijat vaikuttavat niiden onnistumiseen? Miten niitä voidaan parantaa? Näiden kysymysten pohjalta opinnäytetyön tuloksina syntyi opinnäytetyön teoriaosuus, määrittelyn asiakirjapohjat sekä määrittelyn ohjeistus. Kysymyksiä ja tuloksia käsitellään tarkemmin seuraavissa alaluvuissa.

6.1 Opinnäytetyön eteneminen

Alkuperäisten tutkimuskysymysten jälkeen lähdettiin selvittämään ohjelmistomäärittelyn teoriapuolta. Taustatietoa hankittiin mm. alan kirjallisuudesta, havainnoimalla, haastatteluilla ja olemassa olevista määrittelyprojekteista. Taustatiedon tarkoituksena oli tarjota vastauksia kaikkiin opinnäytetyön tutkimuskysymyksiin. Teoriaosuus rakennettiin toimeksiantajan toimintamalliin sopivaksi.

Teoriaosuus auttoi myöhemmin tekemään toimeksiantajan nykytila-analyysin (luku 3), joka tarjosi vastauksia kysymyksiin: Miksi määrittelyprosessi ja -dokumentaatio eivät ole toimivia tällä hetkellä? Mitkä toimijat vaikuttavat niiden onnistumiseen? Teoriaosuuden pohjalta määrittelyprosessin ja -dokumentaation ongelmakohtat olivat melko ilmeisiä ja niiden kartoittaminen helppoa. Kun määrittelyiden nykytila ja ongelmakohtat olivat selvillä voitiin siirtyä seuraavan, ehkä tärkeimpään kysymykseen: miten määrittelyprosessia ja -dokumentaatiota voidaan parantaa?

Ongelmakohtien perusteella oli selvää, että ongelmiin tulisi tarttua ainakin laa- timalla määrittelyä helpottavia asiakirjapohjia sekä ohjeistus määrittelyn tekoon. Määrittelyn kehityskohteita lähdettiin kartoittamaan löydettyjen ongelmakohtien pohjalta (luku 4). Kun näiden kehityskohteiden perusteella luotiin vaatimuksia, joita määrittelyn tulisi saavuttaa, oli tavoitetila koossa.

Tarvittavia toimia tavoitetilan saavuttamiseksi alettiin pohtia luvussa 5. Jo aiemmin oli selvinnyt, että määrittelyn tilaa lähdetään parantamaan asiakirjapohjilla ja ohjeistuksella, joten luvussa viisi (5) sovellettiin näitä keinoja tavoitetilan vaatimusten saavuttamiseksi. Asiakirjapohjia ja määrittelyn ohjeistusta hiottiin koko opinnäytetyön tekemisen ajan.

6.2 Työn teoriaosuus

Opinnäytetyön teoriaosuuden tarkoitus oli tarjota perusteita opinnäytetyön aikana tehtäville kehitystoimille. Teoriaosuuden teko aloitettiin työn alkuvaiheessa ja sen rooli opinnäytetyön etenemisessä oli suuri. Ilman sitä ei olisi voinut tehdä luotettavia jatkotoimia, kuten nykytila-analyysiä tai tavoitetilan selvittämistä. Opinnäytetyön teoriaosuutta voidaan käyttää osana toimeksiantajan perehdytysmateriaalia henkilöstöhallinnan työkalujen puolen määrittelyn työtehtäviin sekä tarjoamaan vanhoille työntekijöille lisätietoa määrittelyprosessista ja hyvistä toimintatavoista.

6.3 Toteutetut asiakirjapohjat

Opinnäytetyön tuloksina syntyi seuraavat asiakirjapohjat: konseptisuunnitelma (liite 4), käyttötapaukset (liite 1) sekä vaatimusmäärittely (liite 5). Konseptisuunnitelma perustuu toimeksiantajalla jo olemassa olevien asiakasprojektien konseptisuunnitelmien sisältämiin tietoihin. Käyttötapaukset-asiakirjan käyttötapauksen kuvausmalli perustuu Wiegerson (2003, 142) kirjassaan kuvailemaan malliin. Vaatimusmäärittely-asiakirjan sisällysluettelo pohjautuu Wikipedian (2011) Ohjelmiston vaatimusmäärittely -artikkeliin.

Valmiita vaihtoehtoja asiakirjapohjiksi on olemassa lukematon määrä. Luodut asiakirjapohjat on valittu toimeksiantajan projekteihin sopivuuden perusteella. Asiakirjapohjat on luotu ajatellen laajoja asiakasprojekteja ja niitä voi vapaasti muokata sopimaan jokaiseen projektiin.

6.4 Toteutettu ohjeistus

Opinnäytetyöhön liittyen toteutettiin määrittelyn ohjeistus henkilöstöhallinnan työkalujen asiakasprojekteille (liite 3). Ohjeistuksen tarkoituksena on tarjota tuotantotiimille suunta hyviin käytänteisiin. Ohjeistuksesta selviää tiimin vastuualueet, tikettikäytännöt sekä dokumentoinnin ja versionhallinnan suositellut toimintatavat. Ohjeistuksen tarkoituksena ei ole pakottaa tiettyyn kaavaan, vaan auttaa dokumentoinnin ja käytänteiden yhtenäistämässä. Ohjeistus voi myös toimia perehdytysmateriaalina uusille työntekijöille.

Ohjeistuksen sisältö koottiin tunnistettujen kehityskohteiden ja tavoitetilan avulla. Ohjeella pyritään tarttumaan mahdollisimman moneen ongelma-kohtaan. Ohjeiden laadinnassa otettiin huomioon muiden tiimin jäsenien toivomukset siitä, miten määrittelyä tulisi tulevaisuudessa tehdä, dokumentoida ja ylläpitää.

7 POHDINTA

Opinnäytetyön tavoitteena oli saada työn toimeksiantajan henkilöstöhallinnan työkalujen asiakasprojektien määrittelydokumentaatiosta kustannustehokkaampi ja hallittavampi kokonaisuus. Mielestäni tavoite onnistuttiin saavuttamaan. Määrittelyprosessi ja -dokumentaatio on nyt selkeä kokonaisuus, selkeine vaiheineen ja tekijöineen. Opinnäytetyön tuloksina syntyneitä asiakirjapohjia on jo alettua hyödyntämään uusissa alkavissa asiakasprojekteissa. Laaditun ohjeistuksen perusteella on alettu järjestämään vaatimusten ja määrittelyiden saata- vuutta paremmaksi.

Verrattuna tilanteeseen ennen opinnäytetyön tekoa ovat määrittelyn tarkoitus, vaiheet ja sijainti asiakasprojektissa paljon selkeämmät. Nyt asiakasprojektin alkaessa on selvää koska määrittely aloitetaan, miten se aloitetaan, miten se etenee ja koska se lopetetaan. Lisäksi määrittelyn dokumentointiin on olemassa valmiit asiakirjapohjat ja ohjeet valmiiden määrittelyiden tallennuksesta sekä ylläpidosta.

Todellinen hyödyllisyys kustannuksien suhteen tulee selviämään vasta pidem- män ajan jälkeen, kun uusilla toimintamalleilla ja asiakirjapohjilla on saatu toteu- tettua asiakasprojekteja ylläpitovaiheeseen asti. Opinnäytetyön kehitystoimin- nan tulosten on tarkoitus vaikuttaa myös ylläpitovaiheeseen ajallisina ja rahalli- sina säästöinä. Tämän opinnäytetyön rajoissa ei kuitenkaan voida toteuttaa kat- tavaa arviointia kehitystoiminnan lopullisista tuloksista.

Opinnäytetyön tarkoituksena oli selvittää nykyisen määrittelyprosessin ja -dokumentaation ongelmakohtia. Jo työn teon alkaessa oli kehittynyt kuva siitä, mitkä ovat tällä hetkellä suurimmat ongelmat määrittelyssä. Mielestäni ongel- makohdat saatiin onnistuneesti selvitettyä ja ratkaistua tehdyillä kehitystoimilla. Sitä, ilmeneekö uuden toimintamallin kautta uusia ongelmakohtia, on vielä liian aikaista sanoa. Toimintamallin hyödyt ja mahdolliset haitat selviävät vain ajan kuluessa.

Seuraavana toimenpiteenä voisi olla moduulien perusversioiden dokumentaatioiden kehittäminen. Ihannetilanne olisi, jos uuden asiakasprojektin alkaessa määrittelyn voisi aloittaa moduulin perusversion dokumentaatiosta. Perusversion dokumentaatio sisältäisi käyttötapaukset ja sanalliset vaatimusmäärittelyn, joiden pohjalta määrittelyitä voitaisiin alkaa räätälöimään asiakkaalle sopivaksi. Tämä opinnäytetyö ja sen tulokset tarjoavat hyvän pohjan jatkokehitykselle, johon toivon toimeksiantajani ryhtyvän.

LÄHTEET

Alexander, I., Stevens, R. 2002. Writing Better Requirements. Addison-Wesley Professional.

Boehm, B. W., Papaccio, P. N. 1988. Understanding and Controlling Software Costs. IEEE Transactions on Software Engineering 14 (10), 1462–1477. Luettu 28.07.2011.
http://faculty.ksu.edu.sa/ghazy/Cost_MSc/R6.pdf

Grady, R. B. 1999. An Economic Release Decision Model: Insights into Software Project Management. In Proceedings of the Applications of Software measurement Conference, 227–239. Orange Park, FL: Software Quality Engineering.

Gül, H. 2008. SWOT Analysis of Technical Education and the Evaluation of its Effectiveness. World Applied Sciences Journal 4, 45-50. Luettu 09.11.2011.
http://akademikpersonel.kocaeli.edu.tr/hgul/sci/hgul05.07.2010_09.06.37sci.pdf

Hayes, B. 2003. A Lucid Interval. Reprint. American Scientist the magazine of Sigma X, the Scientific Research Society 91 (6), 484. Luettu 10.08.2011.
<http://www.cs.utep.edu/interval-comp/hayes.pdf>

IEEE. 1990. IEEE Std 610.12-1990: "IEEE Standard Glossary of Software Engineering Terminology." Los Alamitos, CA: IEEE Computer Society Press.

Kelly, J. C., Sherif, J. S. & Hops, J. 1992. "An Analysis of Defect Densities Found During Software Inspections." Journal of Systems and Software 17 (2), 111–117.

Leffingwell, D. 1997. Calculating the Return on Investment from more Effective Requirements Management. American Programmer 10 (4), 13–16.

Rinzler, B. 2009. Telling Stories: A Short Path to Writing Better Software Requirements. John Wiley & Sons. Luettu 12.08.2011.
<http://proquest.safaribooksonline.com.elib.tamk.fi/book/software-engineering-and-development/software-requirements/9780470437001/firstchapter>

Robertson, S., Robertson, J. 2010. Mastering the requirements process. 2nd edition. London N.J. : Addison-Wesley.

Wieggers, K. E. 2003. Software requirements : practical techniques for gathering and managing requirements throughout the product development cycle. 2nd edition. Redmond (Wash.) : Microsoft Press, cop.

Wikipedia. 2011. Ohjelmiston vaatimusmäärittely. Luettu 20.05.2011.
http://fi.wikipedia.org/wiki/Ohjelmiston_vaatimusmäärittely

Käyttötapaukset

Projektin nimi

Asiakkaan nimi

Laatijan nimi

Luomispvm

(jatkuu)

SISÄLTÖ

JOHDANTO	3
TOIMIJAT	4
KÄYTTÖTAPAUS 1 (TEE TILAUS)	5
KÄYTTÖTAPAUS 2	6

(jatkuu)

JOHDANTO

Johdanto luvussa on tarkoitus lyhyesti kuvailla mistä dokumentissa on kyse. Kuvataan miten käyttötapaukset on valittu ja millä periaatteella ne on priorisoitu. Tämä dokumentti on vain ohje käyttötapauksen laadintaan ja voit vapaasti muokata sitä sopimaan projektiisi.

Jokaisella käyttötapauksella pitää olla tunnistetiedot. Tietojen tulisi sisältää ainakin uniikin käyttötapaus ID:n, nimen, tekijän, muokkaajan, luomispäivän ja muokauspäivän. Muuta tarpeelliset tiedot selviävät esimerkiksi käyttötapauksessa kohdassa Käyttötapaus 1 (Tee tilaus). Esimerkiksi käyttötapaukseen on otettu verkkokauppa.

Huom! Jotta käyttötapauksia voi alkaa laatimaan, on oltava tiedossa projektin toimijat ja ainakin joitakin vaatimuksia, joilla aloittaa. Prioriteetit käyttötapauksille olisi hyvä määrittää jokaisessa projektissa. Prioriteettien asteet voi itse määrittää. Prioriteetit voivat olla esim. matala, kohtalainen, korkea. Kun alat tekemään käyttötapauksia, lisää niiden eteen otsikko, joka on sama kuin käyttötapauksen nimi.

(jatkuu)

TOIMIJIAT

Toimijat kohdassa on hyvä listata ylös kaikki käyttötapauksiin liittyvät toimijat.

Toimijat
Asiakas
Ylläpito
Tavarantoimittaja
.....

(jatkuu)

KÄYTTÖTAPAUS 1 (TEE TILAUS)

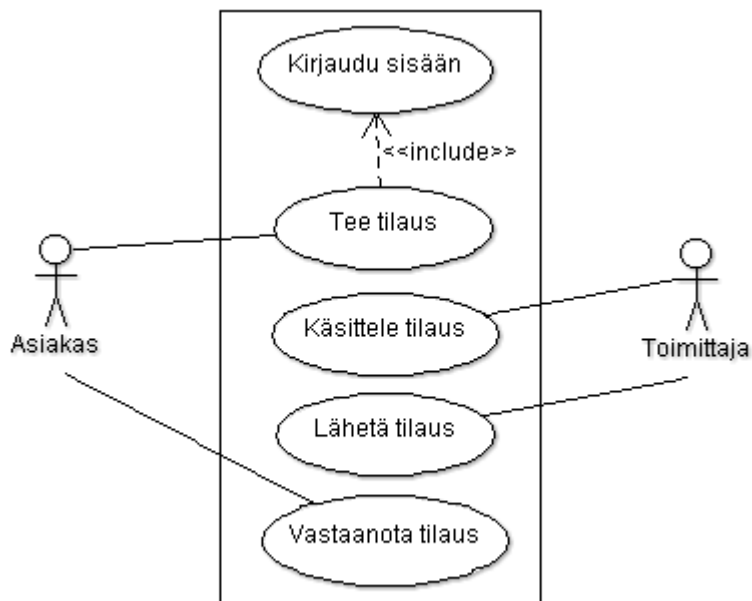
Käyttötapaus ID	KT-1	Käyttötapauksen nimi	Tee tilaus
Luonut	Viivi	Viimeksi muokannut	Teppo
Luotu	19.08.2011	Viimeksi muokattu	20.08.2011
Toimijat	Asiakas		
Kuvaus	Tähän lisätään lyhyt kuvaus tilanteesta, jota käyttötapaus selittää. Esimerkissä se voisi olla: Asiakas löytää verkkokaupasta haluamansa tuotteen ja lisää sen ostoskoriinsa. Hän maksaa ja tilaa tuotteen, verkkokauppa vahvistaa tilauksen.		
Esiehdot	Tässä kuvataan mahdollisia ehtoja, jotka pitää olla täytettynä ennen kuin käyttötapaus aloitetaan ja voidaan viedä loppuun onnistuneesti. Esimerkiksi: <ol style="list-style-type: none"> 1. Asiakas rekisteröityy järjestelmään 2. Asiakkaalla on tunnukset järjestelmään 3. Asiakas on kirjautunut järjestelmään 		
Normaali eteneminen	1.0 Tee tilaus verkkokaupasta <ol style="list-style-type: none"> 1. Asiakas valitsee haluamansa tuotteen 2. Valittu tuote on varastossa 3. Tuote siirtyy ostoskoriin 4. Asiakas maksaa tuotteen 5. Tilaus vahvistetaan 		
Vaihtoehtoinen eteneminen	1.1 Maksa tuote myöhemmin (askeleen 3 jälkeen) <ol style="list-style-type: none"> 1. Asiakas tilaa tuotteen laskulle 2. Tilaus vahvistetaan 		
Poikkeukset	1.0.P.1 Tuotetta ei ole varastossa (askeleen 1 jälkeen) <ol style="list-style-type: none"> 1. Verkkokauppa antaa ilmoituksen "Tuote tilapäisesti loppunut." 2. Verkkokauppa kysyy haluaako hän ilmoituksen sähköpostiinsa, kun tuotetta on taas saatavilla <ol style="list-style-type: none"> 3a. Asiakas valitsee "Kyllä" ja tilaa toisen tuotteen 4a. Asiakas valitsee "Kyllä" ja poistuu verkkokaupasta 5a. Asiakas valitsee "Ei" ja tilaa toisen tuotteen 6a. Asiakas valitsee "Ei" ja poistuu verkkokaupasta 		
Sisältää	Käyttötapaus voi sisältää toisia käyttötappauksia. Esimerkkikäyttötapaus voisi sisältää käyttötapauksen nimeltään "KT-10 Asiakas rekisteröityy järjestelmään".		
Käyttöiheys	Arvioitu käyttöiheys. Esimerkiksi "Kerran kuukaudessa jokaista asiakasta kohden."		
Prioriteetti	Korkea		
Lopputulokset	Asiakas on onnistuneesti tilannut tuotteen ja vastaanottaa sen postitse viikon kuluessa.		
Huomioitavaa	1. Selvitettävä kuinka pitkä eräpäivä laskulle annetaan, oltava selvillä 23.08.2011 mennessä.		

(jatkuu)

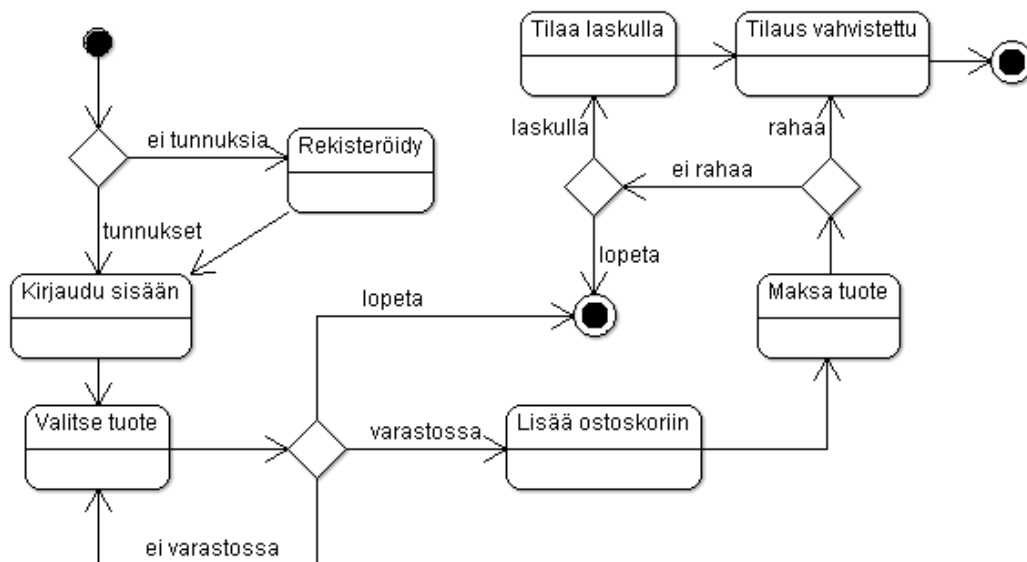
KÄYTTÖTAPAUS 2

Käyttötapaus ID	Käyttötapauksen nimi
Luonut	Viimeksi muokannut
Luotu	Viimeksi muokattu
Toimijat	
Kuvaus	
Esiehdot	
Normaali eteneminen	
Vaihtoehtoinen eteneminen	
Poikkeukset	
Sisältää	
Käyttöiheys	
Prioriteetti	
Lopputulokset	
Huomioitavaa	

Käyttötapauskaavio



Toimintokaavio



Asiakasprojektin
vaatimusmäärittelyn
dokumentoinnin ohjeistus

Viivi Hildén

2011

(jatkuu)

SISÄLTÖ

1 DOKUMENTTIEN LAADINTA	3
1.1 RACI taulukko.....	3
2 MÄÄRITTELYIDEN SAATAVUUS	7
3 MUUTOKSET JA UUDET OMINAISUUDET	8
4 TIKETIT	9
5 DOKUMENTTIEN HALLINTA	10
5.1 Arkistointi.....	10
5.2 Versionhallinta	11

(jatkuu)

1 DOKUMENTTIEN LAADINTA

Tämän dokumentin tarkoituksena on tarjota ohjeistus uusille ja vanhoille työntekijöille asiakasprojektien toiminnallisen määrittelyn ja dokumentoinnin käytänteihin. Tarkempaa tietoa itse määrittelyprosessista löytyy Määrittelyprosessin kehittäminen opinnäytetyöstä luvusta 2 (Hildén 2011).

Projektien määrittelydokumentaatiot tulisi laatia, joko projektipäällikön ja/tai määrittelijän toimesta. Tilanteissa, joissa on mahdollista laatia määrittelyt useamman hengen toimesta on se erittäin suositeltavaa. Kaikista dokumenteista tulee selvittää dokumentin luoja ja mahdolliset muokkauspäivät, sekä muokkaaja (ks. kohta 5.2 Versionhallinta).

Jokaisesta asiakasprojektista tulisi laatia ainakin konseptisuunnitelma, käyttötapausdokumentti, sekä vaatimusmäärittelydokumentti. Tarkemmat ohjeet dokumenttien sisällöistä löytyvät dokumenttipohjista, jotka löytyvät yhteisestä kansioista Z:\Yhteisia\Lerlin-tiimi\Asiakirjapohjat. Mikäli et pääse kansioon ja tarvitset asiakirjapohjia ota yhteyttä tuotantoon.

1.1 RACI taulukko

Taulukon vaakariveillä on jokin tehtävä tai dokumentti ja sarakkeissa eri toimijat. Raci kirjainten selitteet ovat R = responsible (vastuullinen), A = accountable (vastuussa oleva), C = consulted (neuvoja) ja I = informed (tiedotettava). Jokaisella tehtävällä on oltava ainakin yksi R-henkilö, jokaisella tehtävällä on vain yksi A-henkilö, joka valvoo tehtävän valmistumista, jokaisella tehtävällä voi olla nollasta rajattomaan C-henkilöitä, joilta voidaan kysyä neuvoja ja jokaisella tehtävällä voi olla nollasta rajattomaan I-henkilöitä, joita tiedotetaan tehtävästä.

(jatkuu)

TAULUKKO 1. RACI taulukko toiminnallisen määrittelyiden vastuista.

Dokumentti/ Tehtävä	Projektipäällikkö	Määrittelijä	Kehittäjä	Testaaja	QA
Konseptisuunnitelma	R	C / I	C / I		A
Käyttötapaukset	R / C / I	R	I	I	A
Vaatimusmäärittely	R	R	C / I	I	A
Kaaviot	C / I	R	R / C / I	I	
Vaatimusten ylläpito	R / C / I	R / C / I	C / I	C	A
Tikettien hallinnointi	C / I	R / C / I	R / C / I	R / C / I	A

Projektitiimi koostuu normaalisti projektipäälliköstä, yhdestä tai useammasta määrittelijästä, kehittäjästä ja testaajasta. Lisäksi jokaiselle projektille voidaan erikseen määrittää yksi tai useampi henkilö toimimaan QA:na eli laadunvarmistajana. Laadunvarmistajan tarkoitus on huolehtia, että jokainen dokumentti tulee valmiiksi ja on tehty huolellisesti. Laadunvarmistajan tarve arvioidaan projektikohtaisesti. Mikäli projektilla ei ole erikseen laadunvarmistajaa, siirtyy A (vastuussa oleva) vastuu normaalisti henkilölle, jolla on myös R (vastuullinen) vastuu tehtävästä.

Konseptisuunnitelman laatii normaalisti projektipäällikkö. Määrittelijät ja kehittäjät voivat auttaa vastaamalla kysymyksiin ja heitä informoidaan suunnitelman valmistumisesta ja sisällöstä. Suunnitelman pohjalta tehdään aikataulu ja kustannusarviot. Konseptisuunnitelman sisältö ja tyyli on normaalisti vapaamuotoinen. Yleinen pohja konseptisuunnitelmaan löytyy määrittelydokumenttipohjien joukosta.

Käyttötapaukset laatii joko määrittelijä ja/tai projektipäällikkö yhteistyössä asiakkaan kanssa. Projektipäällikkö vastaa kysymyksiin ja hänet pidetään kehittäjien sekä testaajien kanssa informoituna käyttötapausten sisällöstä ja valmistumisesta. Käyttötapausten kerätään vaatimukset vaatimusmäärittelydokumenttia varten. Mikäli katsotaan, että projekti noudattaa ennalta määritettyjä perusversion rajoja ja prosesseja, voidaan käyttötapaukset jättää laatimatta.

(jatkuu)

Perusversion käyttötapauksia voi verrata olemassa olevien projektien käyttötapauksiin. Käytännössä perusversion käyttötapauksina voi pitää edellisen moduulin kehitystä edistäneen asiakasprojektin käyttötapauksia. Jos projekti vaatii räätälöintiä, on käyttötapaukset suositeltavaa sisällyttää määrittelyihin. Käyttötapausten avulla on helppo varmistaa, että oma ymmärrys prosessin etenemisestä vastaa asiakkaan ajatusta.

Vaatusmäärittelyn laativat määrittelijä ja/tai projektipäällikkö. Kehittäjältä voidaan kysyä neuvoa ja kehittäjä pidetään informoituna testaajan kanssa vaatimusmäärittelyn sisällöstä ja valmistumisesta. Kun vaatimusmäärittely on valmis, se tulee hyväksyttäväksi niin asiakkaalla kuin muulla kehitystiimillä. Kaikkia hyväksymisen jälkeen ilmeneviä tarpeita arvioidaan tapauskohtaisesti (kts. 2 Uudet ominaisuudet).

Määrittelyjä tukevia kaavioita laatii normaalisti määrittelijä ja apua tähän saa projektipäälliköltä sekä kehittäjiltä. Testaajat pidetään informoituna kaavioiden sisällöstä ja valmistumisesta. Kaavioita laaditaan normaalisti helpottamaan kokonaisuuksien hahmottamista ja niiden ei välttämättä tarvitse olla virallisia UML kaavioita, vaan ne voivat olla esimerkiksi fläppitaululle piirrettyjä hahmotelmia. Kaavioita ei ole tarpeen varmistaa laadullisesti, koska ne eivät normaalisti tarjoa uutta tietoa projektiin, vaan tukevat jo tiedossa olevia asioita. UML mallinnusta on mahdollista ja suotavaa käyttää tukemaan projektin määrittelyssä. Hyviä kaavioita ovat käyttötapauskaavio, toimintokaavio ja luokkakaavio. Kehittäjän vastuulla on noudattaa Drupalin suositeltuja käytäntöjä koodin dokumentaatiosta ja rakenteesta.

Vaatumusten keräämisen apuna voidaan käyttää erilaisia prototyyppejä. Kuvia ja piirroksia voi laatia kuka tahansa kehitystiimin jäsen tai asiakkaan edustaja. Asiakkaan pyynnöstä tehdyt muutokset on dokumentoitava ja muutoksen tekijä oltava selvitetävissä. Ilmenevät vaatimukset on dokumentoitava vaatimusmäärittelyyn.

6 (12)

Vaatimusten ylläpidosta vastaa projektipäällikkö ja määrittelijät. Molemmat ovat myös tietoisia toistensa tekemistä muutoksista ja he usein myös kuuluvat projektin jäseniin, jotka on valtuutettu muokkaamaan määrittelyitä alkuperäisen hyväksynnän jälkeen.

Tikettejä hallinnoivat normaalisti määrittelijät, kehittäjät ja testaajat. He ovat henkilöt jotka purkavat vaatimukset tiketeiksi, toteuttavat ne ja testaavat ne. He kaikki voivat antaa tietoa tiketeistä ja ovat myös informoituna niiden tilanteesta. Projektipäällikkö pidetään myös ajan tasalla ja häneltä voidaan pyytää apua.

(jatkuu)

2 MÄÄRITTELYIDEN SAATAVUUS

Kun määrittelyt on hyväksytetty sekä asiakkaalla että tuotantotiimillä, viedään määrittelyt trac intraan. Ei siis liitetä suoraan dokumentaatiota Wikiin. Intraan on vietävä määrittelyt, jotka liittyvät toimintoihin, rooleihin, vaatimuksiin jne., jotka merkittävästi poikkeavat työkalun perusversiosta ja perusprosessista. Tietojen viennissä on käytettävä omaa harkintakykyä, siitä mikä on tarpeellista tietoa. Tietojen riittävyys tulee tarkistaa muulta tuotantotiimiltä.

Tiedot on vietävä seuraavan rakenteen mukaiseen paikkaan intra traciin:

wiki: **HRD / AsiakkaanNimi**

Jokainen asiakasprojekti tulee sijoittaa HRD asiakasprojektien alle. Asiakasprojektit nimetään asiakkaan nimellä.

wiki: **HRD / AsiakkaanNimi / ProjektinNimi**

Mikäli asiakkaalle toteutetaan useampi projekti, voi vanhan dokumentaation nimetä uudelleen ja jakaa dokumentaatiot projektien nimien mukaan omiksi sivuikseen.

(jatkuu)

3 MUUTOKSET JA UUDET OMINAISUUDET

Kun alkuperäinen määrittely on hyväksytetty asiakkaalla ja tuotantotiimillä, tehdään sen hetkisistä määrittelyistä baseline-vaatimukset. Näihin vaatimuksiin verrataan kaikkia tuotteen rakentamisen aloittamisen jälkeen ilmeneviä muutoksia tai uusia ominaisuuksia.

Baseline-dokumentin jälkeen uusia ominaisuuksia eivät saa hyväksyä toteutettavaksi kuin ennalta määrätyt ihmiset. Muutoksen tai uudistuksen laajuuden mukaan arvioidaan mahdollisen palaverin tarvetta. Mikäli kyseessä on iso muutos, palaverissa selvitetään muutoksen vaikutus aikatauluun, budjettiin sekä muihin ominaisuuksiin. Jos ominaisuus hyväksytään, lisätään tarvittavat tiedot määrittelydokumentaatioon ja mahdollinen lisälaskutus otetaan huomioon.

Uusien ominaisuuksien hyväksymistä kannattaa harkita tarkkaan, koska jokainen projekti reagoi eri tavalla uusiin tai muuttuneisiin ominaisuuksiin. Yleisimmin ongelmia ilmenee kuitenkin aikatauluissa ja budjeteissa. Hylätyistä ominaisuuksista ja vaatimuksista on hyvä pitää kirjaa. Jokaisesta hylätystä ominaisuudesta on merkittävä ylös ainakin käsittelypäivä, käsittelijät, sekä hylkäämisen perustelut. Tällä tavalla voidaan varmistaa, ettei sama vaatimus tule uudelleen käsittelyyn samaan projektiin ja myös tulevien projektien kannalta tiedetään, millaisia ominaisuuksia on aikaisemmin hylätty ja miksi.

(jatkuu)

4 TIKETIT

Tikettejä hallinnoidaan Intra tracissä ja Lerlin tracissä.

Intra tracissä löytyvät mm. asiakaskohtaisia asetuksia, asennuksia, teemoja, lokalisoiteja, ohjeistusta jne koskevat tiketit. Myös asiakaskohtaiset dokumentaatiot löytyvät täältä.

Jokainen hyväksytty toiminnallisuus, uusi ominaisuus, korjaus tai muutos, joka tuotteeseen tehdään, on tehtävä tiketin pohjalta. Tällaisia tikettejä hallinnoidaan Lerlin tracissä. Yhdestä tiketistä selviää tiketin luomisaika, moduuli, johon tiketti liittyy, tiketin luoja, tikettiin tehdyt muutokset, tiketin toteuttaja, muutoksen testaaja ja hyväksyjä, sekä hyväksymis ajankohta. Projektin etenemisen seuranta on näin helppoa tracistä saatavien raporttien avulla. Jokaisella tiketillä on sen tilaa kuvaava status. Statukset ovat "new", "assigned", "accepted", "testing", "fixed", "pending", "invalid" tai "won't fix".

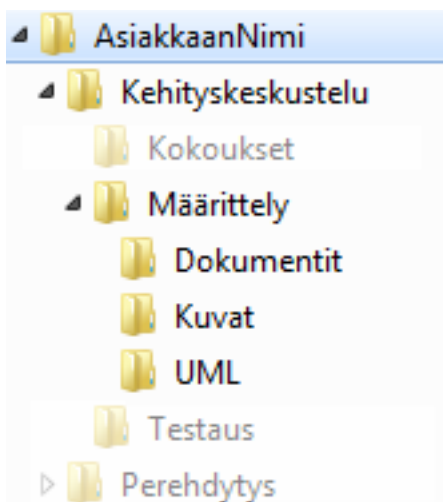
Vaatimukset puretaan tiketeiksi Lerlin traciin ja priorisoidaan tärkeyden mukaan. Prioriteetti määräytyy kiireellisyyden ja vakavuuden mukaan. Tiketteihin, jotka koskevat toteutettavia toiminnallisuuksia, on hyvä sisällyttää linkki Intra tracin sivulle, josta kyseisen toiminnallisuuden määrittely löytyy. Jos vaatimukseen tulee myöhemmin muutoksia, on alkuperäinen tiketti säilytettävä ja merkittävä uudistukset siihen. Lisäksi vaatimusta koskeva dokumentaatio on päivitettävä Intra traciin, kuitenkin niin, että ison muutoksen vaatimuksen alkuperäinen muoto on säilytettävä. Käytännössä tämä tarkoittaa, että dokumentista tehdään uusi versio ja vanha säilytetään arkistoissa. Dokumenttien ja vaatimusten versionhallinta on työntekijöiden omalla vastuulla!

(jatkuu)

5 DOKUMENTTIEN HALLINTA

5.1 Arkistointi

Määrittelyiden ja muiden dokumenttien säilytys on työntekijöiden omalla vastuulla. Jokaiselle projektille on kuitenkin suositeltavaa luoda oma projektihakemistonsa, jossa on selkeästi eroteltuna eri osa-alueiden kansiot. Kansioita luotaessa on hyvä muistaa, että kansiot nimeltään ”Muuta” tai ”Sekalaiset” ovat huono idea. Jokaiselle dokumentille pitää löytyä selkeästi oma paikkansa. Seuraavassa on esimerkki siitä millainen asiakasprojektin hakemistorakenne voisi olla.



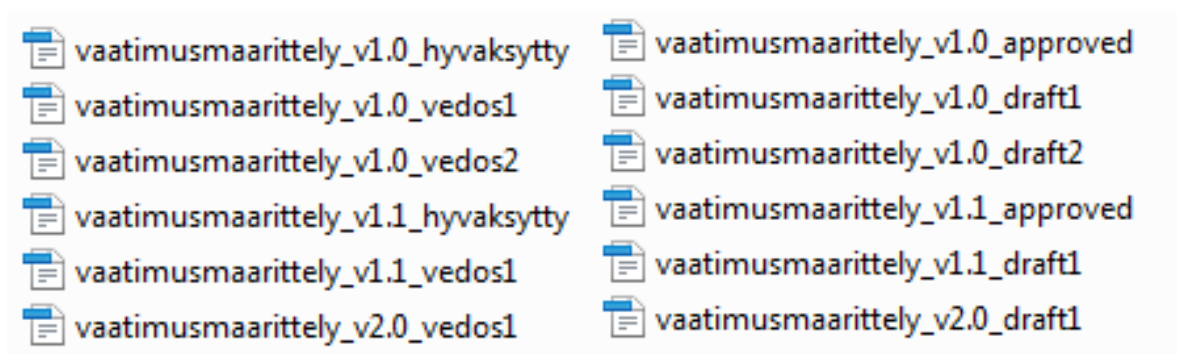
Kuva 5-1. Esimerkki asiakasprojektin hakemistorakenteesta.

Projektihakemisto on hyvä nimetä selkeästi asiakkaan nimellä. Mikäli asiakkaalla on käytössä useampi työkalu/moduuli, voidaan jokaiselle työkalulle luoda oma alikansionsa (esim. kehityskeskustelu ja perehdytys). Määrittely-kansion tulisi sisältää kaikki projektin vaatimukset. Määrittelyn alta tulisi löytyä dokumentit, kuvat ja UML alikansiot. Tarpeen ja määrittelyiden mukaan alikansioita voi luoda lisää, esimerkiksi prototyypeille tai demoille. Muista kuitenkin nimetä kaikki kansiot järkevästi!

(jatkuu)

5.2 Versionhallinta

Dokumenttien versiointi on työntekijän omalla vastuulla. Hyvä versioinnin tapa on esimerkiksi lisätä dokumentin nimen perään versionumero ja vedoksen numero (ks. kuva 5-2). Vedoksien numeroita kasvatetaan joka muutoksen yhteydessä ja kun versio hyväksytään, lisätään versionumeron perään merkintä hyväksyty. Seuraavassa versiossa kasvatetaan joko versionumeroa kokonaisella yksiköllä, tai pienen muutoksen yhteydessä desimaalilla. Jälleen vedosten numeroa kasvatetaan kunnes versio on hyväksyty.



Kuva 5-2. Esimerkki versionhallinnasta.

Dokumenttien alkupäähän on hyvä lisätä versiotaulukko. Lyhyissä dokumenteissa versionhallinnaksi saattaa riittää pelkkä dokumenttiin lisättävä versiotaulukko. Taulukkoon merkitään dokumentin luomis- ja muokkaamispäivät sekä dokumenttien muokkaajat. Taulukkoon voi myös lyhyesti kuvata tehdyt muutokset tai lisäykset, sekä silloisen dokumentin versionumeron. Versioiden hyväksymispäivämäärät voi merkitä myös taulukkoon, tekijä sarakkeeseen tulisi merkitä henkilöt, jotka ovat hyväksyneet version. Seuraavassa esimerkki versiotaulukosta.

(jatkuu)

Taulukko 5-1. Esimerkki versiotaulukosta.

Muutos	Päivämäärä	Tekijä	Versio
Dokumentti luotu	01.05.2011	VH	1.0
Lisätty käyttötapaus KT1	12.05.2011	TT	1.0
Versio 1.0 hyväksytty	12.05.2011	VH, TT, TK	1.0
Muokattu käyttötapaus- ta KT12	17.05.2011	TK	1.1
.....			

Konseptisuunnitelma

Projektin nimi

Asiakkaan nimi

Laatijan nimi

Luomispvm

(jatkuu)

SISÄLTÖ

KOKONAISUUS	3
VAATIMUKSIA.....	3
TOTEUTUKSEN HYÖDYT	3
AIKATAULU.....	3
HUOMIOITA	4
DEMO	4

(jatkuu)

KONSEPTISUUNNITELMA

Projektin nimi

Asiakas

Moduuli(t) (käyttöön otettavien moduulien nimet)

Kokonaisuus

Tässä kuvataan lyhyesti tilanne. Mitä moduuleita asiakas haluaa käyttöön, onko hänellä ennestään jotakin työkaluja käytössään, jos mitä niille tapahtuu.

Vaatimuksia

Kuvataan sanallisesti mitä asiakas toivoo saavuttavansa uuden työkalun avulla.

Toteutuksen hyödyt

Kuvataan mitä hyötyjä uuden työkalun uskotaan tuovan asiakkaan liiketoiminnalle.

Aikataulu

Alustava etenemisaikataulu. Karkea kuvaus viikkojen tai jopa kuukausien tarkkuudella.

(jatkuu)

Huomioita

Onko jotakin erityistä, mitä tulisi ottaa huomioon.

Demo

Mistä löytyy demo ja milloin. Lisäksi käyttäjätunnukset ja salasanat demoan.

VAATIMUSMÄÄRITTELY

ASIAKKAAN NIMI

PROJEKTIN NIMI

TEKIJÄN NIMI

PÄIVÄMÄÄRÄ

(jatkuu)

Sisältö

1 JOHDANTO	3
1.1 Tarkoitus ja kattavuus.....	3
1.2 Tuote ja ympäristö	3
1.3 Määritelmät, termit ja lyhenteet	3
1.4 Yleiskatsaus dokumenttiin	3
2 YLEISKUVAUS	4
2.1 Ympäristö.....	4
2.2 Toiminta	4
2.3 Käyttäjät	4
2.4 Käyttäjädokumentit	4
2.4 Yleiset rajoitteet	4
2.5 Oletukset ja riippuvuudet	5
3 KÄYTTÖTAPAUKSET	5
3.1 Käyttötapauskaavio	5
3.2 Käyttötapaukset.....	5
4 MUUT OMINAISUUDET	5
4.1 Suorituskyky ja vasteajat	6
4.2 Saavutettavuus ("käytettävyys"), toipuminen, turvallisuus, suojaukset	6
4.3 Ylläpidettävyys.....	6
4.4 Siirrettävyys ja yhteensopivuus	6
5 SUUNNITTELURAJOITTEET	6
5.1 Standardit	6
5.2 Ohjelmistorajoitteet	7
5.3 Muut rajoitteet	7
6 JATKOKEHITYSAJATUKSIA	7
LIITTEET	7

(jatkuu)

1 Johdanto

Asiakirjapohja on vapaasti muokattavissa projektikohtaisesti.

1.1 Tarkoitus ja kattavuus

Kuvataan lyhyesti projektin tarkoitus ja yleiset rajat. Tunnistetaan tuote tai tuotteet joita määritellään, myös moduulin versionumero on hyvä mainita.

1.2 Tuote ja ympäristö

Kuvataan tuote ja ympäristö, johon toteutetaan

1.3 Määritelmät, termit ja lyhenteet

Listataan asiakasprojektissa käytettävät tärkeimmät määritelmät, termit, lyhenteet ja niiden selitteet.

1.4 Yleiskatsaus dokumenttiin

Laajemmissa projekteissa selite dokumentin etenemisestä ja sisällöstä.

(jatkuu)

2 Yleiskuvaus

2.1 Ympäristö

Kuvataan, missä ympäristössä tuote tulee toimimaan sekä mahdolliset muut ohjelmat, joiden kanssa tuote joutuu toimimaan yhteen.

2.2 Toiminta

Kuvataan asiakasprojektin perusprosessi, joka tuotteella olisi vietävä läpi.

2.3 Käyttäjät

Tunnistetaan tuotteen tulevat käyttäjät ja heidän roolinsa.

2.4 Käyttäjädokumentit

Luetellaan tuotteen kanssa toimitettavat mahdolliset käyttäjädokumentit, kuten käyttöohjeet, koulutusmateriaalit ja verkosta löytyvät ohjeet.

2.4 Yleiset rajoitteet

Kuvataan tunnetut rajoitteet tuotteelle.

(jatkuu)

2.5 Oletukset ja riippuvuudet

Kuvataan mahdolliset riippuvuudet muista tuotteista tai ympäristöistä.

3 Käyttötapaukset

3.1 Käyttötapauskaavio

Liitetään suoraan otsakkeen alle, tai halutessa liitteeksi asiakirjan loppuun.

3.2 Käyttötapaukset

Käyttötapaukset voidaan liittää kukin omana alakohtanaan (3.2.1 jne.) tai liitteeksi dokumentin perään (suositeltavaa).

4 Muut ominaisuudet

Kuvataan tarpeen mukaan muita ei-toiminnallisia vaatimuksia.

(jatkuu)

4.1 Suorituskyky ja vasteajat

Kuvattavat suorituskyvyt ja vasteajat on oltava mitattavissa. Vaatimuksista tulisi antaa selkeät aikarajat, joiden puitteissa tuotteen on suoritettava tietyt toiminnot.

4.2 Saavutettavuus ("käytettävyys"), toipuminen, turvallisuus, suojaukset

4.3 Ylläpidettävyys

4.4 Siirrettävyys ja yhteensopivuus

5 Suunnittelurajoitteet

5.1 Standardit

Kuvataan asiakkaan tuotteelle asettamat standardit.

(jatkuu)

5.2 Ohjelmistorajoitteet

Kuvataan mahdolliset ohjelmistorajoitteet.

5.3 Muut rajoitteet

Kuvataan mahdolliset muut rajoitteet.

6 Jatkokehitysajatuksia

Tuotteen tulevaisuuden näkymät, uudet versiot, asiakkaan kehitysideat jne.

LIITTEET