

Jasse Virta

CRM-OHJELMISTON LOKALISOINTITESTAUS
Case: Roihu Solutions Oy

Opinnäytetyö
Tietojenkäsittelyn ko.

Maaliskuu 2017




**Kaakkois-Suomen
ammattikorkeakoulu**

KUVAILULEHTI

 XAMK Kaakkois-Suomen ammattikorkeakoulu	Opinnäytetyön päivämäärä 10.03.2017
Tekijä(t) Jasse Virta	Koulutusohjelma ja suuntautuminen Tietojenkäsittelyn koulutusohjelma
Nimeke CRM-ohjelmiston lokalisointitestaus – Case: Roihu Solutions Oy	
Tiivistelmä <p>Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa CRM-ohjelmiston lokalisointitestaus Roihu Solutions Oy:lle. Lokalisointitestauksen tavoitteena on saada tarjottua ohjelmiston loppukäyttäjälle luonteva tapa kokea ohjelmisto kotimaisena tuotteena sen ulkomaisesta alkuperästä huolimatta.</p> <p>Tarkoituksena oli testata melkein valmista Amazonin pilvipalvelussa toimivaa CRM-ohjelmistoa ja löytää sieltä suomentamattomat osiot sekä muut kielelliset poikkeavuudet ennen tuotteen päätymistä ensimmäiselle pilottiasiakkaalle. Testauksen tarkoitus on saada toimeksiantajalle tieto mahdollisista virheistä, jotta ne voidaan korjata. Tämän seurauksena varmistetaan, että pilotointi onnistuu moitteettomasti ja tätä kautta Roihu Solutionin mahdollisuudet saada uusia asiakkaita kasvavat.</p> <p>Testauksessa löydettyjen virheiden ja niistä tehtyjen virheraporttien avulla opinnäytetyön toimeksiantaja Roihu Solutions Oy on taas askeleen lähempänä lokalisoitua lopputulosta sekä valmiimpana antamaan tuotteen loppukäyttäjien käytettäväksi.</p>	
Asiasanat (avainsanat) CRM, ohjelmistotestaus, lokalisointitestaus	
Sivumäärä 31	Kieli Suomi
Huomautus (huomautukset liitteistä)	
Ohjaavan opettajan nimi Jukka Selin	Opinnäytetyön toimeksiantaja Roihu Solutions Oy

DESCRIPTION

	Date of the bachelor's thesis 10 March 2017
Author(s) Jasse Virta	Degree programme and option Business Information Technology
Name of the bachelor's thesis CRM software localization testing - Case: Roihu Solutions Oy	
Abstract <p>The subject of this thesis was to design and implement localization tests of the CRM software for Roihu Solutions Oy. The aim for localization testing was to provide the end users of the software with a natural experience of a domestic product regardless of its origin.</p> <p>The purpose was to test almost completed CRM software running on Amazon cloud services and to find translation errors before the software ends up with the first customer. Goal of the testing was to provide Roihu Solutions with the necessary information about possible errors for correction. This was done to ensure successful first customer experience and improve the company's possibilities to reach new clients. Detected and reported errors will help Roihu Solutions to get closer to a localized product and to be more ready to provide end users with this product.</p>	
Subject headings, (keywords) CRM, software testing, localization	
Pages 31	Language Finnish
Remarks, notes on appendices	
Tutor Jukka Selin	Bachelor's thesis assigned by Roihu Solutions Oy

SISÄLTÖ

1	JOHDANTO	1
2	CRM.....	2
2.1	CRM yleisesti	2
2.2	CRM:n haasteet & tavoitteet	4
2.3	CRM:n osa-alueet	8
2.4	Erilaisten CRM-ohjelmistojen vertailua	11
3	OHJELMISTOTESTAUS.....	13
3.1	Ohjelmiston elinkaari.....	13
3.2	Ohjelmistotestaus yleisesti.....	15
3.3	Testausprosessi	18
3.4	Testaustavat	21
4	CRM-OHJELMISTON LOKALISOINTITESTAUS	25
4.1	Testaussuunnitelma.....	26
4.2	Testaus	28
5	PÄÄTÄNTÖ	31
	LÄHTEET	32

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on toteuttaa ja suunnitella CRM-ohjelmiston lokalisoititestaus, jotta se saataisiin ensimmäiselle pilottiasiakkaalle käyttöön mahdollisimman virheettömänä. CRM-ohjelmiston on tarkoitus tuottaa lisäarvoa sen käyttönottaville yrityksille ja heidän asiakkailleen.

Käyn opinnäytetyössä läpi CRM:ää yleisesti, sekä sen erilaisia käyttöönottostrategioita. Tämän jälkeen tutkin ohjelmistotestausta ja erilaisia testaustapoja sekä käyn läpi lyhyesti ohjelmiston elinkaaren. Opinnäytetyön ensisijaisena tavoitteena on löytää mahdolliset ohjelmiston käänkövirheet. Tätä kautta saadaan toimitettua CRM-ohjelmiston ensimmäiselle pilottiasiakkaalle mahdollisimman hyvin lokalisoitu lopputuote.

Luvussa 2 käyn läpi CRM:ää yleisesti, tarkastelen CRM:n vaikutusta yritysten ja heidän asiakkaidensa kannalta. Tutkin ja käyn läpi tilastollisesti suurimmat CRM-ohjelmistotoimittajat ja alan markkinajohtajat maailmanlaajuisesti. Luvussa käyn läpi myös yleisesti CRM:n eri haasteita ja sille asetettuja tavoitteita, sekä CRM:n eri osat alueet. Lopuksi vertailen yleisimpiä CRM-ohjelmistoja niiden käytettävyyden ja hinnan perusteella.

Luvussa 3 selvitän ohjelmiston elinkaaren lyhyesti ja siinä yleisimmin käytetyn vaihejakomallin. Tutkin ohjelmistotestausta yleisesti ja sen erilaisia työvaiheita. Käyn läpi testausprosessin V-mallin mukaisesti ja esittelen ohjelmistotestauksessa yleisimmin käytetyt testaustavat.

Luvussa 4 laadin testaussuunnitelman, jonka yhteydessä tulee myös ilmi, kuinka toteutan CRM-ohjelmiston lokalisoititestauksen ja minkälaisia menetelmiä käytän. Tämän jälkeen esittelen CRM-ohjelmiston testausprosessin ja käyn läpi testauksessa löytämiäni virheitä.

2 CRM

Customer Relationship Management, joka on nykyään yksi olennaisimpia käsitteitä yritystoiminnassa tänä päivänä. Se sisältää asiakaslähtöisen ajattelutavan yrityksessä, kuin myös siihen yhteydessä olevat IT-järjestelmät.

Asiakkuudenhallinnan ajatuksena on tuoda lisäarvoa liiketoimintaa harjoittavalle yritykselle, sekä heidän asiakkailleen. Hyvin toteutetun asiakkuudenhallinnan hyödyt yrityksessä ovat kulujen vähentäminen ja kohdentaminen asiakashankinnassa, hyvä asiakaspysyvyys sekä kilpailukyvyn parantaminen.

2.1 CRM yleisesti

CRM tulee sanoista Customer Relationship Management. Se otti paikkansa ja vakiintui kielenkäyttöön vasta 1900-luvun lopulla. Suomessa se on vielä niin tuore termi, että suomen kielessä sille ei ole vielä vakiintunut selvää suomennosta. Alalla käytetään sellaisia käännöksiä kuten asiakashallinta, asiakassuhteen johtaminen ja asiakkuuksien johtaminen (Oksanen 2010, 21). Tämän opinnäytetyön teoriaosuuksissa käytetään sanaa asiakkuudenhallinta tai lyhennystä CRM. Vaikka sana asiakkuudenhallinta antaa ymmärtää, että kysymyksessä olisi yksittäisen asiakkuuden hallinta, niin näin ei kuitenkaan ole vaan se sisältää organisaation kaikkien asiakkuuksien hallinnan (Mäntyneva 2001, 9).

Vielä tänäkin päivänä tuntuu siltä, että markkinoinnin eri perusvariaatioiden, eli toisin sanoen hinnan, tuotteen, jakelun sekä markkinointiviestinnän ohella organisaation on syytä keskittää toiminta-ajatustaan yhä enemmän asiakaslähtöistä suuntaa kohden. Käytännöllisesti katsoen se tarkoittaa sitä, että asiakkaiden vaatimukset havaitaan ja otetaan huomioon, jotta niitä voidaan toteuttaa entistä paremmin. Toisaalta asiakkaat eivät ole valmiita maksamaan tällaisesta toimenpiteestä yhtään ylimääräistä. Sen takia organisaation täytyy löytää keinot sellaiseen ratkaisuun, josta ei koidu asiakkaille turhia lisäkustannuksia (Mäntyneva 2001, 10).

Asiakkuudenhallinta on monipuolinen termi, se voidaan ajatella käsitteenä toimintatavoille ja tietojärjestelmille, joilla yritys systemaattisesti hallinnoi asiakkuuksiaan. Prosessina, joilla hallinnoidaan jokaista asiakaskohtaamista, kuten esimerkiksi myyntiä,

markkinointia ja asiakaspalvelua. Lähestymistapana asiakkaiden hankkimiseen ja tunnistamiseen. Tietojärjestelmänä liiketoiminnassa, jonka tarkoitus on auttaa suunnittelemaan, aikatauluttamaan sekä johtamaan asiakaspalvelun, myynnin ja markkinoinnin toimia. Liiketoimintastrategiana, jolla tarkastellaan asiakkaiden kannattavuus, tuotto ja tyytyväisyys suurimman mahdollisen voiton maksimoimiseksi (Oksanen 2010, 22).

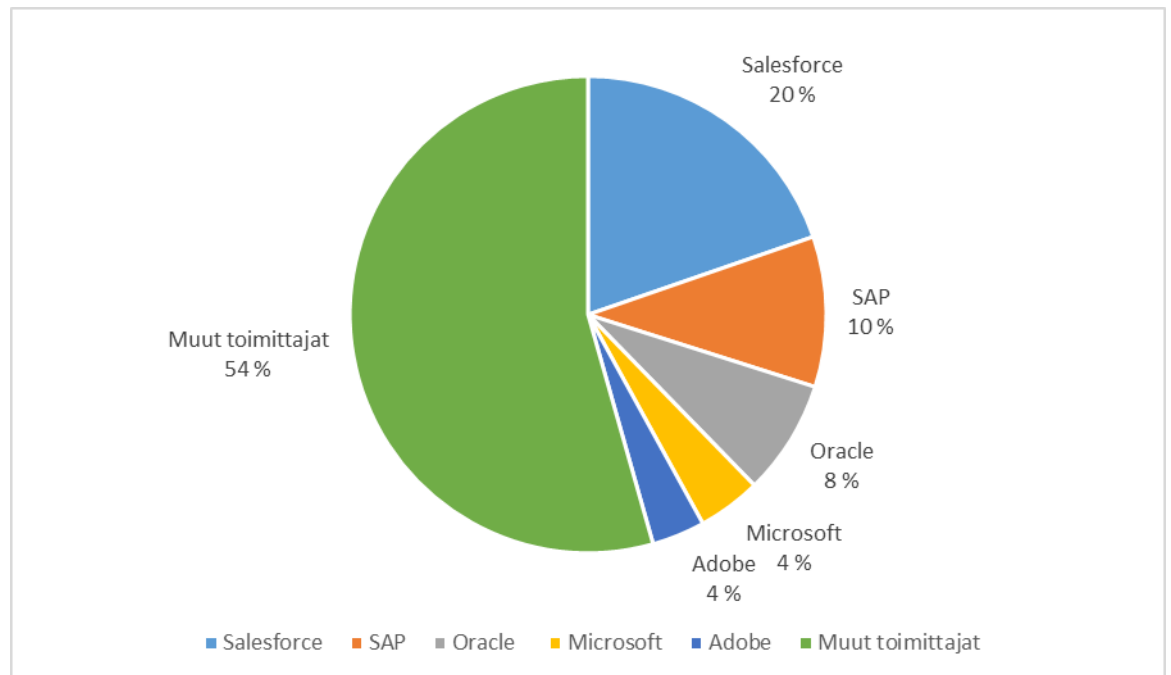
Vuonna 2013 tehdyssä tutkimuksessa ilmeni, että Suomessa käytettiin edelleenkin Excel-taulukkolaskentaohjelmistoa CRM:n korvikkeena. Peräti yksi viidestä suomalaisesta liikevaihdoltaan yli 50 miljoonan euron yrityksestä toimi ilman CRM-järjestelmää. Liikevaihdoltaan 20-50 miljoonan euron yrityksissä CRM puuttui noin puolelta (Tivi 2013).

Globaalisti CRM-ohjelmistojen tarve näkyy yritysten panostuksina CRM-ohjelmistoihin. Maailmanlaajuisesti myynti kasvoi vuonna 2015 yhteensä 12,3 prosenttia 26,3 miljardiin dollariin, edellisen vuoden 23,4 miljardista dollarista, Tämä käy ilmi taulukosta 1. Vuosikatsauksessa SaaS Mallisissa eli pilvessä toimivilla CRM ratkaisuilla, Oli 27 prosentin siivu kaikkien CRM-järjestelmien liikevaihdosta, näin sanoo talouslehti Forbes.

Yritys	2015	2015	2014	2015
	Tuotto	Markkinaosuus (%)	Tuotto	Markkinaosuus (%)
Salesforce	5,170.9	19,7	4,268.5	18,2
SAP	2,684.4	10,2	2,669.0	13,0
Oracle	2,046.5	7,8	2,119.0	9,1
Microsoft	1,141.5	4,3	951.1	4,1
Adobe	936.8	3,6	738.1	3,2
Muut	14,307.7	54,4	12,658.3	55,4
Yhteensä	26,287.8	100%	23,404.0	100%

TAULUKKO 1. CRM-ohjelmistojen myynti 2014-2015 (Gartner 2016, mukailen)

CRM-ohjelmistotoimittajien viiden kärki hallitsi 45 prosenttia globaaleista markkinoista vuonna 2015. Suurin toimittaja oli Salesforce, toiseksi suurin SAP, kolmantena Oracle, neljäntenä Microsoft. Muiden toimittajien markkinaosuudet olivat yhteensä 54,4 prosenttia (Forbes 2016.) Havainnollistaminen kuvassa 1.



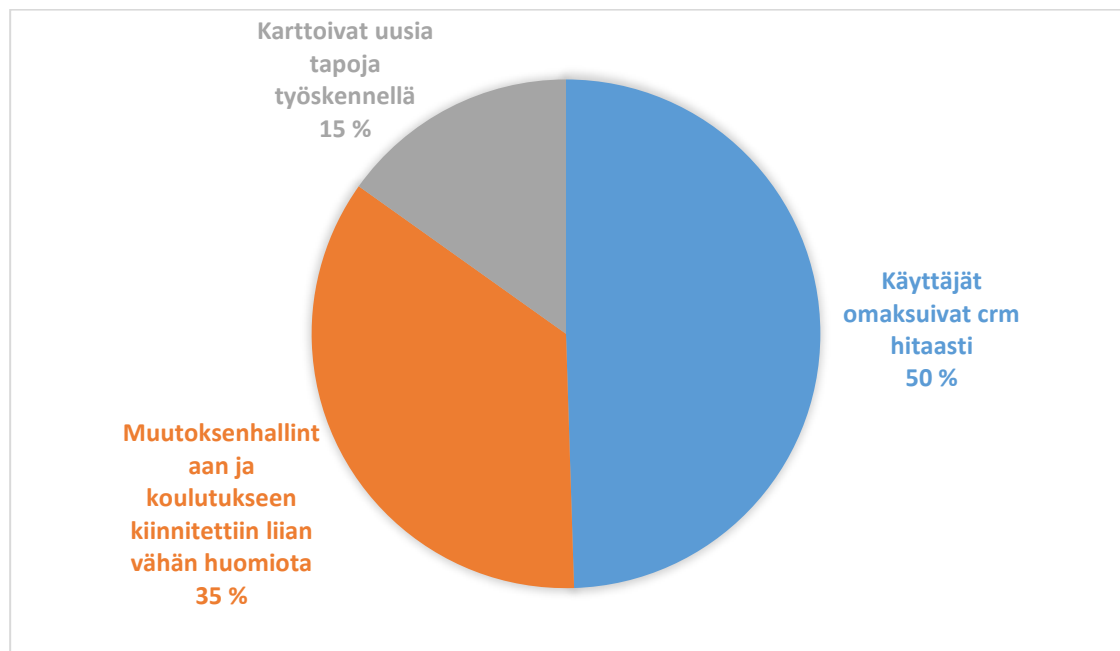
KUVA 1. Suurimmat CRM-ohjelmistotoimittajat vuosina 2014-2015 (Forbes 2016, mukaillen)

Lopulta CRM hanke ei useimmiten ole mutkaton projekti. Jotta CRM saavuttaisi sille asetetut tavoitteet, yrityksen on pystyttävä vakuuttamaan johto kuin myös henkilöstö CRM:n tarpeellisuudesta. Tämän jälkeen täytyy tutkia liiketoimintaprosessit ja tehdä päätös muutostarpeesta. Seuraavaksi yrityksen on ratkaistava, mitkä asiakastiedot ovat heille tärkeitä ja kuinka niitä käytetään. Tämä voi kestää joistakin viikoista vuoteen, riippuen tietenkin yrityksen suuruudesta ja liiketoiminnan monimutkaisuudesta. Nykypäivänäkin useat yritykset kuluttavat satoja tuhansia, elleivät peräti miljoonia asiakkuudenhallintajärjestelmien ohjelmistolisensseihin (Myynti20 2012).

2.2 CRM:n haasteet & tavoitteet

Edesmennyt italialainen organisaatioteoreetikko Claudio Ciborra kavereineen tutki CRM:n etenemistä IBM:ssä yhdeksänkymmentä luvun lopulla. Tutkimukset osoittivat sen edenneen ainoastaan vain joihinkin osiin yritystä. Tutkimuksen eräs keskeisimmistä havainnoista oli, että työntekijät omaksuivat kyllä CRM:n käsitteenä, mutta työskentelivät silti vanhanaikaisilla totutuilla menetelmillä (Oksanen 2010, 25). Opin näytetyön kirjoittajan mielestä, tämä kuvaa erittäin hyvin suurta ongelmaa yritystoiminnassa, joka on muutoksen pelko ja uusien asioiden omaksumisen vaikeus.

Vuonna 2009 Forrester teki tutkimuksen, jossa oli mukana 133 eri organisaatiota eri puolilta maailmaa. Tutkimuksesta kävi ilmi, että kaikista CRM käyttöönotoissa ilmoitetuista ongelmista 22% koskivat niin sanottuja henkilökunnan ongelmia. Kun tutkimuksessa analysoitiin asiaa tarkemmin, todettiin että merkittävimpiä uhkia olivat, että käyttäjät omaksuivat CRM:n hitaasti, muutoksenhallintaan ja koulutukseen kiinnitettiin liian vähän huomiota ja osalla oli vaikeuksia siirtyä vanhoista uusiin toimintatapoihin. Kuvassa 2 merkittävimmät ongelmat prosentuaalisesti (Forrester 2009).



KUVA 2. Henkilökunnan ongelmat CRM käyttöönotossa (Superoffice 2016, Mukailten)

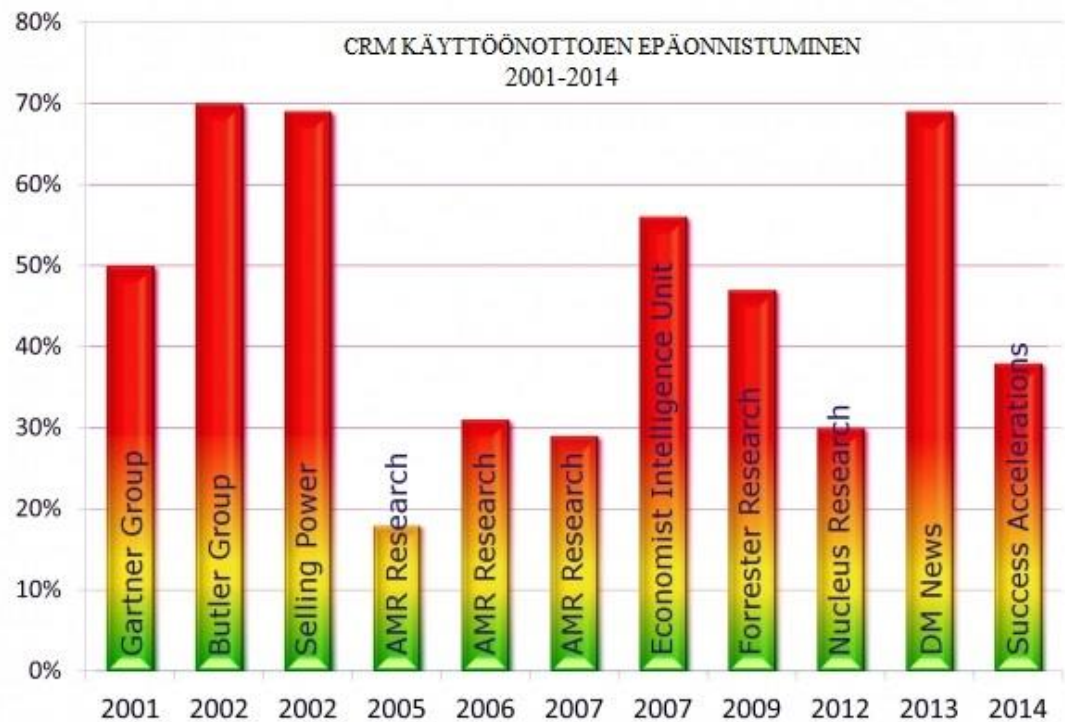
CRM:n puuttuminen ei tavallisesti saa suuria vahinkoja aikaan yrityksessä lyhyellä aikavälillä. Mikäli CRM-järjestelmä lopettaa toimintansa esimerkiksi teknisten ongelmien takia, yrityksen liiketoiminta monesti ainoastaan hidastuu. Silloin kuin

CRM:ää ei käytetä, johtuu se usein muista seikoista kuin teknisistä ongelmista. CRM-järjestelmä on siis olemassa, mutta kukaan ei vain käytä sitä (Oksanen 2010, 25-26).

Ongelmana voidaan pitää myös yleisesti sitä, että alkuinnostuksen jälkeen CRM-järjestelmän käyttö yrityksessä vähenee. Aluksi siitä ei ole suurta haittaa, sillä hyvä myyjä ei tarvitse välttämättä tietojärjestelmää muistaakseen mitä asiakkaiden kanssa on sovittu. Järjestelmän käyttämättömyyden tuomat ongelmat havaitaan usein vasta henkilöstövaihdosten yhteydessä (Tiirikainen 2010, 85). Tästä seuraa CRM-järjestelmien toinen tyypillinen ongelma, että yleensä yhden yksikön tiedonvälitykseen keskittyvässä käyttöönotossa CRM-järjestelmä on tyypillisesti leimattu epäonnistumaan. (Oksanen 2010, 26).

CRM-järjestelmien toteuttamisessa on toki myös muitakin haasteita. Yleensä ongelmat alkavat kohtuuttomista kustannuksista CRM-järjestelmien käyttöönotoissa. Yritykselle on suuri investointi ylläpitää asiakasrekisteriä. Lisäkustannuksia tuovat muun muassa laitteisto, ohjelmisto ja henkilökunta joita tarvitaan ylläpitämään sitä. Joskus nämä kohtuuttomat kustannukset voivat vaikuttaa negatiivisin seurauksin yritykseen, joka voi johtaa CRM:n käyttöönoton epäonnistumiseen (Google 2016).

Koska CRM-järjestelmien käyttöönotossa haasteita on lukematon määrä, niin opinäytetyön kirjoittajan mielestä tässä käytiin läpi niistä tärkeimmät. Kirjoittaja haluaisi vielä lopuksi havainnollistaa CRM:n käyttöönottojen haasteet yhdellä kuvalla (kuva 3). Vuosina 2001-2014 lukuisat tutkimukset ovat osoittaneet yhden yhtenäisen piirteen CRM-järjestelmien käyttöönotossa ja se on epäonnistuminen (Ecaminc 2015).



KUVA 3. CRM:n epäonnistumiset vuosina 2001-2014 (Ecaminc 2015, Mukailen)

Asiakkuuksien korostaminen tarkoittaa asioiden läpi käymistä asiakkaan perspektiivistä. Markkinoinnin oleellinen tavoite on pyrkiä rakentamaan sellaista asiakasarvoa, jota asiakas pitää myös itse tärkeänä. Tämän ohella asiakassuhde tulee pystyä luomaan asiakaskannattavuus säilyttäen. CRM:n tarkoitus on tehostaa organisaation päätöksentekoa niiltä osin, mitä asiakkuuksien kehittämistä organisaatio haluaa jatkossa parantaa (Mäntyneva 2001, 11).

Yleisesti CRM:n tavoite on saada asiakkaat ostamaan omalta yritykseltä, eikä kilpailijalta. Mäntynevan mukaan CRM:n hyödyt voidaan jaotella kahteen ryhmään. Ensimmäiseksi CRM nostaa yrityksen asiakkuuksiin sisältyvää tietoa ja käsitystä siitä, miksi he ostavat. Toiseksi CRM tehostaa myyntiä, markkinointia ja vaikuttavuutta, mikä taas on merkityksellistä markkinoinnin kokonaiskannattavuuden kannalta. (Mäntyneva 2001, 12). CRM:n käyttäminen on tänä päivänä yksi menestyvien myyntiorganisaatioiden avaintekijöitä. Kun myyntiprosessi ja myyntiä edeltävät toimintatavat on tehty niin että myynti CRM ohitse on mahdotonta. CRM:n käyttöaste on tällaisissa organisaatioissa hyvin laadukkaalla tasolla (Zapps 2014).

Oksasen mukaan CRM:n käyttöönoton tavoitteet on saavutettu, jos kahden vuoden päästä käyttöönoton jälkeen seuraavat kriteerit ovat täyttyneet. Ensimmäisenä kriteerinä on, että neljä viidestä suunnitelluista käyttäjästä käyttää järjestelmää. Tämän onnistuminen on helposti mitattavissa. Parhaimmissa tapauksissa käyttäjien määrä on jopa kasvanut. Toisena kriteerinä on että CRM:ää käyttävä organisaatio on riippuvainen CRM-järjestelmästä. Kolmantena kriteerinä on, että CRM-järjestelmä on korvaamattoman tärkeä osa organisaation tiedonhallintaa. Toisen ja kolmannen kriteerien tavoitteet ovat saavutettu, jos CRM-järjestelmä poistettaisiin asiakkaan käytöstä koituisi siitä suuria ongelmia liiketoiminnalle (Oksanen 2010, 29).

2.3 CRM:n osa-alueet

CRM-järjestelmät jaetaan niiden ominaisuuksien perusteella. Yleisesti puhutaan CRM:n eri tyypeistä, jotka ovat strateginen, operatiivinen, analyyttinen ja kollaboraatiivinen. Jokainen edellä mainituista hyödyntää eri ominaisuuksia (Tutorialspoint 2016). Kuvassa 4 havainnollistus.



KUVA 4. CRM osa-alueet (Tutorialspoint 2016, mukailen)

Strateginen CRM on asiakaskeskeinen toimintamalli, jossa yritys asettaa asiakkaan arvon tärkeimmäksi. Tämän toimintamallin periaate on saada potentiaaliset asiakkaat

ostamaan juuri omalta yritykseltä. Tuottamalla ja toimittamalla parempia palveluita suhteessa kustannuksiin kuin kilpailijat (Buttle 2008, 4).

Näin kerätään, erotellaan ja sovelletaan tietoa asiakkaista, jotta saadaan enemmän lisäarvoa asiakkaalle. Tätä kautta yritys oppii jatkuvasti asiakkaista ja heidän vaatimuksistaan. Yritys pystyy näin oppimaan ja sopeutumaan heidän tarpeisiin saaden tätä kautta etulyöntiaseman kilpailussa. Jos yritys ei harkitse tämän tyyppistä CRM toimintamallia, niin se saattaa menettää markkinaosuutta niille yrityksille, jotka käyttävät tätä toimintamallia (Tutorialspoint 2016).

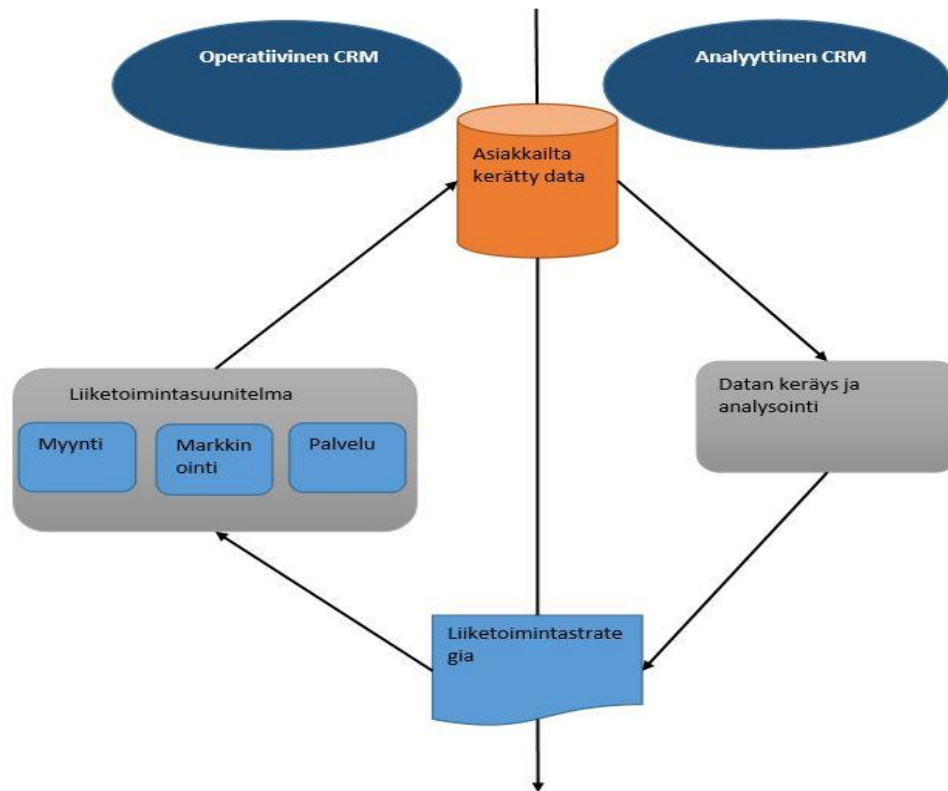
Operatiivisella CRM:lla tarkoitetaan kaikkea mahdollista vuorovaikutusta asiakkaan kanssa. Operatiivisen CRM:n tarkoitus on tehostaa liiketoiminnan eri prosesseja. Prosesseja jotka pitää sisällään myynnin, markkinoinnin ja palveluiden automatisoinnin. Myynnin automaatio auttaa yritystä tehostamaan myyntiprosessia. Tämän kautta saadaan asiakkaalle tarjottua parempaa palvelua joka vaikuttaa asiakaspysyvyyteen ja asiakassuhteeseen.

Markkinoinnin automaation päätarkoitus on selvittää paras tapa tarjota tuotetta ja lähestyä potentiaalisia asiakkaita. Tämän tarkoitus on tehostaa markkinointia tarjoamalla asiakkaille juuri sitä mitä he tarvitsevat. Markkinoinnin automatisointi CRM:ssa voi koostua myös eri moduuleista, kuten kampanjanhallinnasta, joka auttaa yritystä suunnittelemaan ja päättämään viestintäkanavat, joita ovat esimerkiksi sähköpostit ja sosiaalinen media.

Palvelun automaatio mahdollistaa yrityksen säilyttää asiakkuuksiaan tarjoamalla asiakkaille parasta mahdollista palvelua, joka auttaa hyvän asiakassuhteen rakentamisessa. Tämä toimii käytännössä niin että yrityksen puhelinpalveluissa yhteyshenkilönä toimivan ruudulle yritetään saada kaikki tiedot soittajan asiakkuuden historiasta (Techonestop 2016.)

CRM-kokonaisuuden pohjana toimii analyyttinen CRM. Tällä tarkoitetaan itse asiassa työvälineiden takana olevaa dataa, ymmärrystä markkinoista ja omasta asiakaskunnasta. Analyyttinen CRM elää siis vuorovaikutuksessa operatiivisen CRM:n kanssa, sillä se käyttää operatiivisesta CRM:sta saatua arvokasta tietoa polttoaineenaan. Nimensä

mukaan kerätty tieto analysoidaan ja näin liiketoimintaa voidaan kehittää yhä enemmän asiakaslähtöistä suuntaa kohten (Tivi 2006).



KUVA 5. Operatiivinen & Analyttinen prosessi (Techonestop 2016, Mukailten)

Onko sitten operatiivisen ja analyttisen CRM:n välillä eroja. Operatiivinen CRM siis käsittelee myynnin, markkinoinnin ja palvelun automaation, jossa ollaan suorassa vuorovaikutuksessa asiakkaan kanssa ja yritetään vastata asiakkaan vaatimuksiin. Analyttinen CRM taas käsittelee ja analysoi tästä saatua dataa, eikä ole suoraan tekemisissä asiakkaan kanssa (Techonestop 2016).

Kollaboratiivisella CRM:lla tarkoitetaan usein sitä, että erilaiset yhteydenpitotavat asiakkaisiin päin integroidaan. Tämä pitää sisällään järjestelmiä jotka antavat mahdollisuuden hallita asiakkaiden kanssa käytyä kommunikaatiota. Puhelinpalvelu muuttuukin yhteyskeskukseksi, jolla hallitaan esimerkiksi fakseja, tekstiviestejä ja sähköpostikyselyitä. Yritys saa näin enemmän tietoa asiakkaan palvelusta ja tarpeista (Tivi 2006).

Organisaation eri osastot kuten myynti, tekninen tuki ja markkinointi jakavat tietoa joka on kerätty asiakkaalta. Esimerkiksi tekninen tuki kerää asiakaspalautteen, jonka jälkeen jakaa tiedon eteenpäin markkinointiosastolle. Markkinointiosasto saa tiedon tuotteista ja palveluista, jotka kiinnostavat asiakasta. Tarkoituksena yhteistyöllä on parantaa asiakaspalvelun laatua ja tätä kautta lisäämään asiakkaiden tyytyväisyyttä ja uskollisuutta (Searchsaleforce 2016).

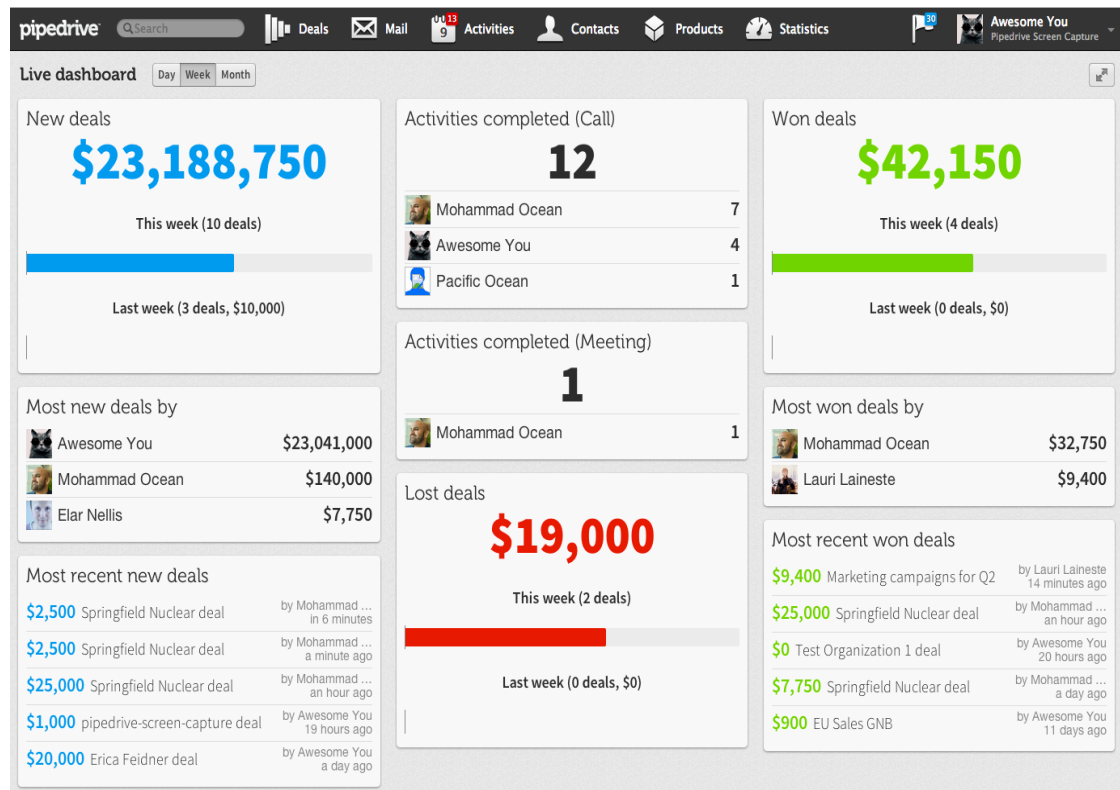
2.4 Erilaisten CRM-ohjelmistojen vertailua

CRM-järjestelmää valittaessa kannattaa kiinnittää huomiota kahteen asiaan. Ensimmäkin on otettava huomioon omat tarpeet, eli pitääkö etusijalla järjestelmän helppoutta vai muokattavuutta. Niin kuin aina, mitä enemmän vaatimuksia, sen korkeampi hinta. Toiseksi valintaa tehdessä olisi päätettävä, keiden suurten teknologiayrityksen väliltä valitsisi. Yleensä nämä vaihtoehdot ovat joko Google tai Microsoft (Vinkkisolusion 2016).

Yleisesti toimintamalleja on kahdenlaisia. Asennettavat järjestelmät ja SaaS-pohjaiset ratkaisut, eli pilvessä toimivat palvelut. Asennusmallisessa toimintatavassa ohjelmisto ostetaan yritykselle ja asennetaan sen omaan infrastruktuuriin. Tämän toimintatavan etuina voidaan pitää asiakastietojen säilyttämistä omilla palvelimilla, sekä räätälöinnin monipuolisuutta omiin tarkoituksiin. Haittoina mainittakoon työläämpi käyttöönotto ja kalliimpi hinta. SaaS-pohjaisissa ratkaisuissa järjestelmä asiakastietoineen ovat sen toimittavan yrityksen palvelimilla, joka toimii internetin kautta pilvipalveluna. Etuina käyttöönoton helppous ja palvelua voidaan käyttää kokonaan ilman oman yrityksen tietohallintoa. Etuina mainittakoon myös se, että järjestelmää voidaan käyttää mistä tahansa, milloin tahansa. Haittoina mainittakoon tiedon säilyttämisen palvelun tarjoajalla sekä räätälöinnin vaikeus omiin tarpeisiin (Ampiiri 2014).

Vinkki Solutions CRM-vertailun mukaan järjestelmät luokitellaan kahteen ryhmään. Helppokäyttöisiin ja vaativiin CRM-järjestelmiin. Helppokäyttöisiä CRM-järjestelmiä ovat Pipedrive, Highrise, Zoho CRM, Base CRM, Insightly jotka sopivat niin pienille kuin keskisuurille yrityksille. Vaativia CRM-järjestelmiä ovat Sugar CRM, SalesForce, Microsoft Dynamics, SuperOffice jotka ovat järkevä valinta isoille yrityksille niiden kattavan räätälöinnin vuoksi (Vinkkisolusion 2016).

Helppokäyttöisissä keskisuurille yrityksille tarkoitetuissa CRM-järjestelmissä käyttäjien mielestä paras ratkaisu oli Pipedrive CRM-järjestelmä. Vaativien ja suurille yrityksille tarkoitettujen järjestelmien parhaana ratkaisuna pidettiin Salesforce CRM-järjestelmää. Tutkimus perustuu g2crowd sivuston käyttäjien arviointeihin (G2crowd 2016).



KUVA 6. Kuvankaappaus Pipedrive CRM-järjestelmän käyttöliittymästä (Technologyadvice 2016)

Vinkkisolusionin tehdyssä hintavertailussa käy ilmi, että helppokäyttöisistä CRM-järjestelmistä kallein vaihtoehto on Base CRM. Highrise luottaa hinnoittelussa varsin edulliseen ratkaisuun. Pipedrive, Zoho, Insightly ovat hinnoiltaan samat yli 10 hengen yrityksille. Vaativien CRM-järjestelmien osalta SugarCRM ja SuperOffice olivat edullisimmat vaihtoehdot, kun taas Salesforce ja Microsoft Dynamics olivat kalleimmat vaihtoehdot (Vinkkisolusions, 2016.) Seuraavassa kuvassa ylläolevien CRM-järjestelmien hinnat kuukausittain käyttäjämäärän mukaan. Taulukko 2 hinnat on ilmoitettu Yhdysvaltain dollareina.

CRM	1	3	5	10	20	30	50	100	200	500
Helppokäyttöiset										
Highrise	24	24	24	49	99	99	99	99	99	99
Zoho	0	0	0	0	240	360	600	1200	2400	6000
Insightly	0	36	60	120	240	360	600	1200	2400	6000
PipeDrive	12	36	60	120	240	360	600	1200	2400	6000
Base CRM	25	75	125	750	1500	2250	3750	7500	15000	37500
Vaativat										
SugarCRM	150	400	400	400	800	1200	2000	4000	8000	20000
SuperOffice	42	126	210	420	840	1260	2100	4200	8400	21000
SalesForce	25	75	125	650	1300	1950	3250	6500	13000	32500
MS Dynamics	325	325	325	650	1300	1950	3250	6500	13000	32500

TAULUKKO 2. CRM-järjestelmien hinnasto käyttäjämäärän mukaan kuukausittain (Vinkkisolutions 2016, Mukailten)

Lopuksi voisin hieman kieliposkella todeta, että CRM-ohjelmiston valinta on riippuvainen useista asioista. Ei ole olemassakaan selkeää ykkösvalintaa. Valintaa tehdessä on syytä tutkia ja selvittää omat tarpeet perinpohjaisesti.

3 OHJELMISTOTESTAUS

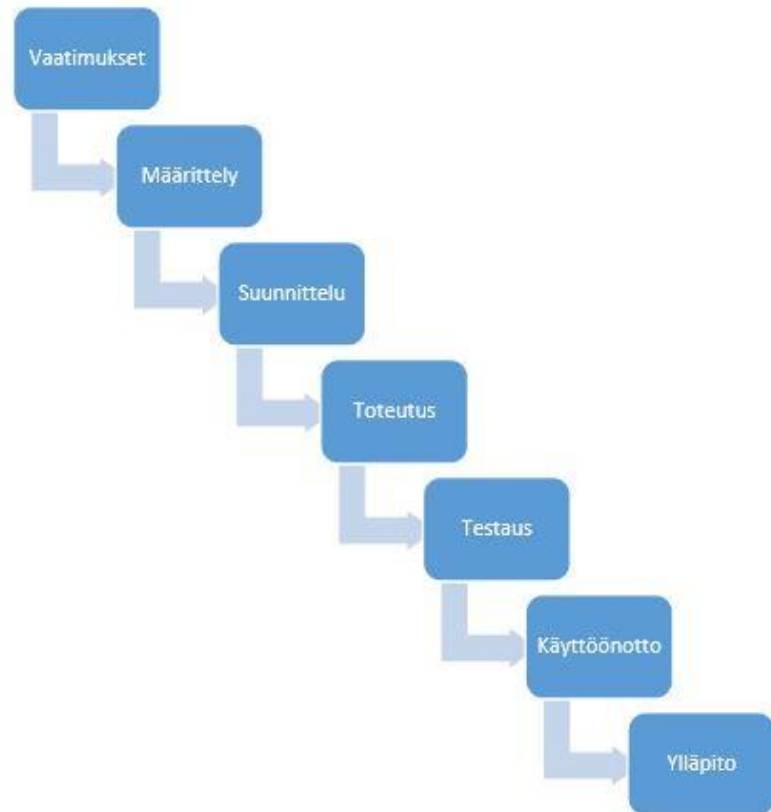
Ohjelmiston testaus on prosessi, jonka tehtävä on varmistaa kohtaako ohjelmiston tulokset sille asetetut vaatimukset. Ohjelmistotestauksen päätarkoitus on varmistaa, että ohjelmisto on virheetön. Yleensä ohjelmissa on virheitä, joita kutsutaan bugeiksi. Käytännössä bugien syntymistä on mahdoton estää ja harva niistä pystytään löytämään ennen ohjelman suorittamista.

3.1 Ohjelmiston elinkaari

Ennen kuin siirrytään ohjelmistotestauksen pariin, on hyvä ymmärtää ohjelmiston elinkaari. Näin saadaan kattavampi ymmärrys ohjelmistotestauksesta ja sitä edeltävistä tapahtumista. Ohjelmiston elinkaari osoittaa sen aikaa joka alkaa sen kehitystyöstä ja loppuu ohjelmiston käytöstä poistoon.

Yleisin tapa jakaa ohjelmiston elinkaari on käyttää vaihejakomallia eli niin sanottua vesiputousmallia (kuva 7). Ohjelmistoprojekti sisältää seuraavat asiat kyseisessä jär-

jestyksessä vaatimusanalyysi, määrittely, ohjelmistosuunnittelu, toteutus, testaus ja lopuksi käyttöönoton ja ylläpidon. Ketterissä kehitysmenetelmissä näitä vaiheita toistetaan, kunnes on päästy haluttuun lopputulokseen. Kaikki vaiheet sisältävät laadunvarmistustoimenpiteitä esimerkiksi tarkastuksia ja testausta. Ohjelmaa tarkastellessa ja testaamalla pyritään siitä poistamaan virheitä ohjelman tekemisen aikaisessa vaiheessa.



KUVA 7. Vesiputousmalli (Testingfreak 2017, Mukailen)

Vaatimusanalyysillä pyritään osoittamaan järjestelmältä odotettavia palveluita ja toiminnalle asetettuja rajoitteita. Vaatimusanalyysi tehdään yhteistyössä ohjelmiston asiakkaan kanssa. Näistä saadaan aikaan ohjelmistovaatimukset, mitä kautta määritellään toteutettava järjestelmä. Tästä tehtyä dokumenttia kutsutaan toiminnalliseksi määrittelyksi (Haikala & Märijärvi 2004, 36-39.)

Suunnitteluvaiheessa suunnitellaan, miten ohjelmisto toteutetaan. Suunnittelun yhteydessä muutetaan vaatimusmäärittelyssä esitetyt vaatimukset ja toiminnot tekniselle

kielelle. Suunnitteluvaihe pitää sisällään ohjelmiston, käyttöliittymän, tietokannan, testauksen ja käyttöönoton suunnittelun (Joensuu 2001).

Toteutusvaiheessa päätetään millä ohjelmointikielellä ohjelmisto toteutetaan. Tämän jälkeen ohjelmointi aloitetaan ohjelmoimalla järjestelmän eri komponentit. Toteutusvaiheessa aloitetaan testaus yksikkötestauksella, josta kerrotaan tarkemmin luvussa 3.3 (Joensuu 2001).

Käyttöönottovaiheessa valmis ohjelmisto otetaan käyttöön. Nimeä käyttöönotto käytetään, kun tuotetta ei julkaista vaan se on tehty tietyn asiakkaan käyttöön. Tämä sisältää asennusvalmiin ohjelmistokokonaisuuden jossa myös yleensä huolehditaan ohjelman asennuksesta asiakkaan kanssa sovittuihin laitteisiin (Xtract 2017).

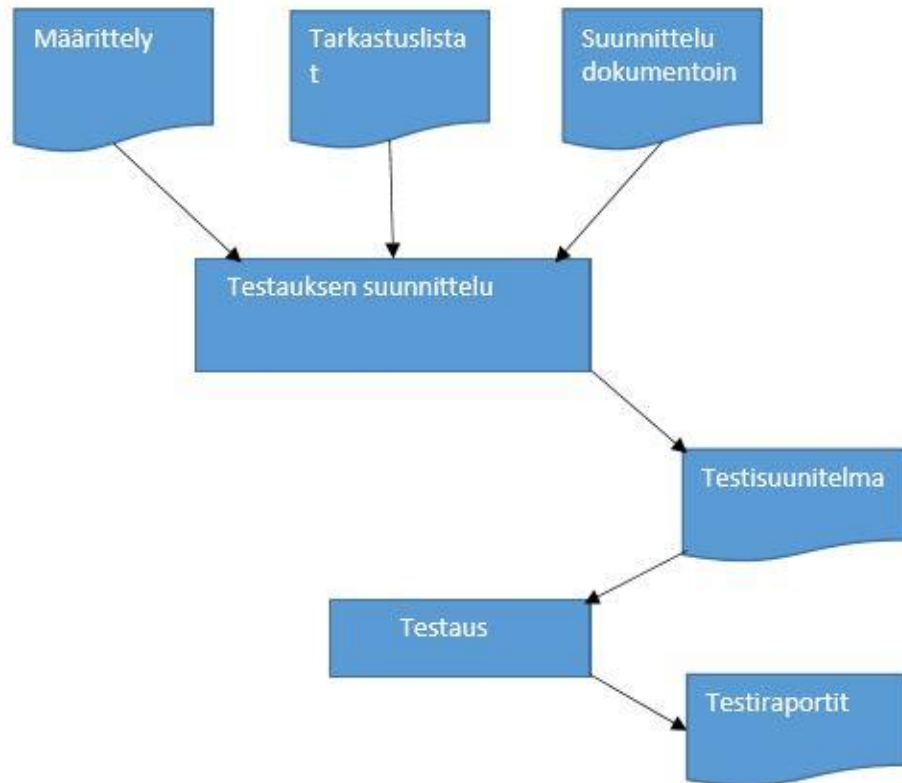
Ylläpito on ohjelman ylläpitämistä, jolla tarkoitetaan asiakkaan ongelmien ratkomista, virheiden korjaamista ja vaatimusten muuttuessa ohjelman muuntamista asiakkaan toiveiden mukaiseksi. Yksinkertaisesti ajateltuna ylläpito voitaisiin jakaa korjaavaan ja täydentävään ylläpitoon. Korjaavassa ylläpidossa ohjelman virheitä pyritään korjaamaan. Täydentävässä ylläpidossa ohjelmaan lisätään toimintoja asiakkaan vaatimuksien perusteella (Haikala & Märijärvi 2004, 41). Ohjelman elinkaari loppuu, kun ohjelmisto poistetaan käytöstä.

3.2 Ohjelmistotestaus yleisesti

Testaus koostuu useista työvaiheista. Työvaiheista kuten suunnittelu, testiympäristön luonti, testin ajo ja tulosten tarkastelu. Näihin työvaiheisiin ja ohjelmointivirheiden löytämiseen sekä korjaamiseen kuluu yleensä yli puolet koko ohjelmistoprojektiin suunnitelluista resursseista.

Puhekielessä sanalla testaus ymmärretään sen tarkoittavan kokeilemistä. Tietojenkäsittelytieteessä puhutaan ohjelmistotestauksesta, joka tarkoittaa yleisesti suunnitelmallista virheiden etsimistä, joko itse ohjelmaa tai sen osaa testaamalla. Tänä päivänäkin testaus määritellään laajemmaksi kokonaisuudeksi kuin ennen, tätä tarkastellaan laadun mittaamisen näkökulmasta. Tämä siksi että testauksen katsotaan sisältävän ne kaikki menetelmät, joilla pyritään parantamaan ohjelmiston laatua (Haikala & Märijärvi 2004, 283-285.)

Ennen testausta tehdään suunnitelma, kuinka testaus etenee ja ne dokumentoidaan. Testauksen edetessä dokumentteja syntyy paljon, sillä jokaisesta suoritetusta testistä tehdään erikseen raportti testin kulusta. Testauksen suunnittelu näytetään kuvassa 8.

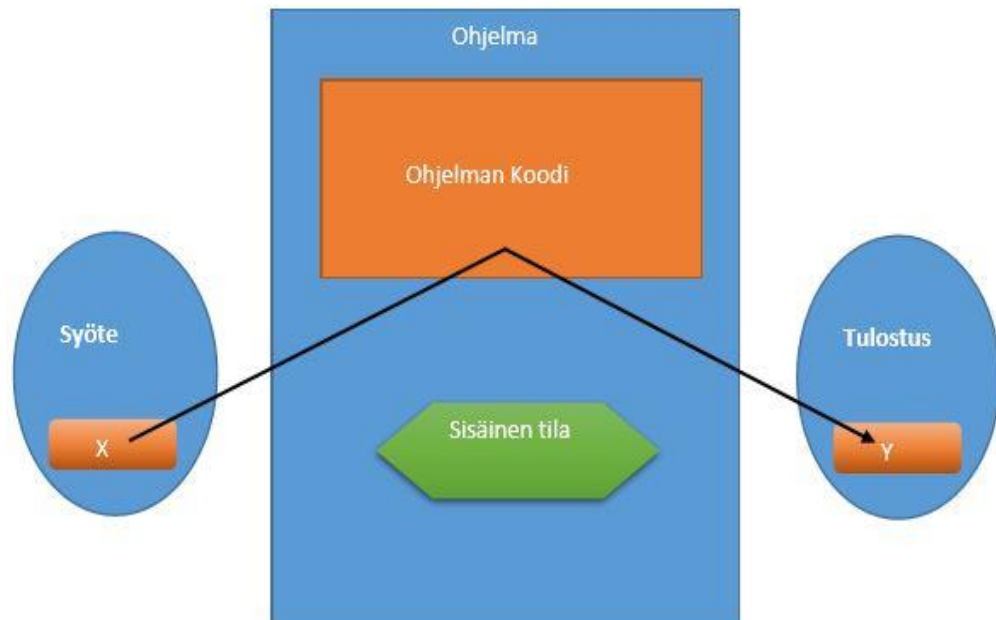


KUVA 8. Testauksen suunnittelu (Haikala & Märijärvi 2004, 298. Mukaillen)

Testaussuunnitelma tehdään sitä varten, että siitä on helppo havainnoida mitä ohjelman osaa testataan ja milloin. Siitä selviää myös millaisia lopputuloksia testauksella olisi tarkoitus saavuttaa (Haikala & Märijärvi 2004, 299). Hyvällä testaussuunnitelmalla varmistetaan testauksen tehokkuus.

Yksinkertaista testaustilannetta voidaan kuvailla seuraavasti (kuva 9). Koodari Henkka tekee ohjelman, ohjelma on tässä tapauksessa valuuttamuunnin, jonka täytyy muuttaa eurot rupliksi. Ohjelmalle syötetään euromäärä, jonka halutaan tulostuvan rupliksi. Ajatellaan euro X:na ja rupla Y:na. Testaaja Tomi testaa ohjelman, jotta testi onnistuu se edellyttää, että tulos Y on validi ja että ohjelman sisäinen tila on muuttunut oikein. Testaustulos riippuu muustakin kuin sille annetusta syötteestä, kuten järjestelmän si-

säisestä tilasta. Sisäisellä tilalla tarkoitetaan esimerkiksi ohjelman muuttujien arvoja (Haikala & Märijärvi 2004, 285).



KUVA 9. Ohjelmiston testaus (Haikala & Märijärvi 2004, 284, Mukailleen)

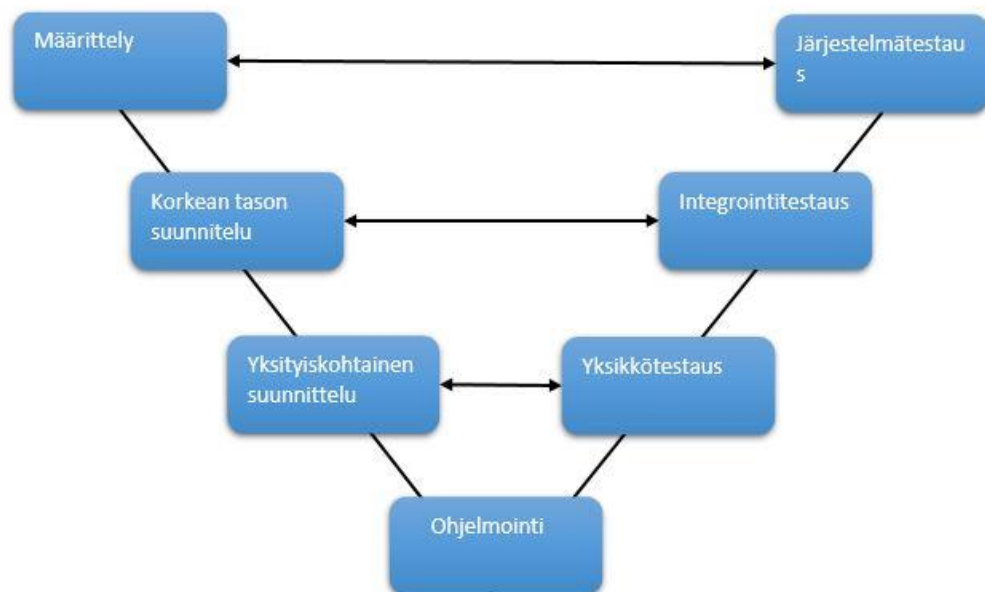
Olenaisempana osana ohjelmistotestauksessa on virheilmoitus (error, bug). Bugi nimitystä käytetään usein tietojenkäsittelytieteessä. Nimi bugi juontaa juurensa vuoteen 1947, jolloin yhdysvaltalainen tietojenkäsittelytieteilijä Grace Murray Hopper kirjasi ensimmäisen tietokonebugin muistiinpanoihinsa, kun hän työskenteli Harvard Mark II sähkömekaanisen laskukoneen kanssa. Koneen ongelma jäljitettiin, ongelmaksi todettiin koneen ja releen väliin juuttunut koiperhonen. Hopper kirjoitti lokikirjaansa ”First actual case of bug being found” (suom. ensimmäinen todellinen tietokonebugi on löydetty). Bugi tulee siis englanninkielisestä sanasta ”bug” joka tarkoittaa suomeksi ötökkää (Computerhistory 2017).

Virhe on poikkeama ohjelmistokoodissa josta yleensä seuraa virheilmoitus ja näin ohjelma ei ehkä toimi kunnolla. Sen vakavuus voi vaihdella käyttäjää ärsyttävästä pienestä ohjelmointivirheestä, tai kokonaan ohjelman käytön estävästä virheestä.

Yleensä ohjelmistovirheitä arvioidaan olevan noin yksi, paria kymmentä ohjelmariviä kohden. Vaikka ohjelma olisi ollut käytössä jo pitkään, arvioidaan siinä olevan yksi ohjelmointivirhe tuhatta riviä koodia kohden. Haikalan ja Märijärven mukaan ohjelmistovirheistä 5% ei edes koskaan löydetä (Haikala & Märijärvi 2004, 287.)

3.3 Testausprosessi

Ohjelmistotestauksessa V-malli edustaa kehitysprosessia. Sitä voidaan pitää vesiputouksmallin laajenuksena, joka on tunnetuin v-malli. Sen sijaan että prosessit kuvattaisiin menevän alaspäin lineaarisesti, prosessit kuvataan alkavan alhaalta ja taipuvan eri suuntiin ylöspäin jolloin siitä muodostuu tyypillinen V-muoto. V-malli kuvaa kehityksen ja testauksen välistä suhdetta, prosessin elinkaaren aikana (kuva 10) (Forsberg & Mooz 1991, 4-5).



KUVA 10. V-Malli (Smarteducation 2017, Mukailten)

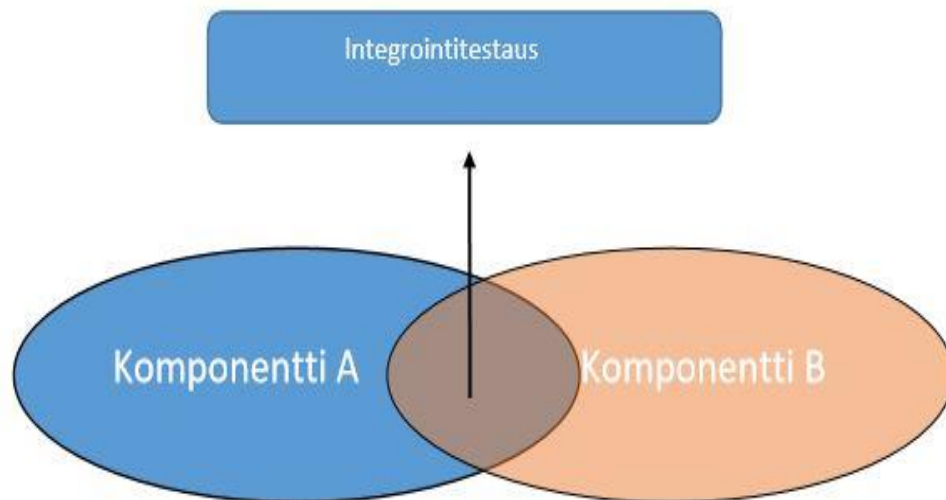
Yleisesti kuvattuna v-malli sisältää yksikkötestauksen (käytetään myös nimeä moduulitestaus), integrointitestauksen ja järjestelmätestauksen. Joissakin tapauksissa järjes-

telmätestausta varten joudutaan tekemään erillinen kenttätestaus tai hyväksymistestaus riippuen ohjelmiston vaatimuksista (Haikala & Märijärvi 2004, 288).

Yksikkötestauksen tavoite on pyrkiä testaamaan pienintä mahdollista ohjelman osaa. Yksikkö on yleensä noin 100-1000 koodiriviä pitkä (Haikala & Märijärvi 2004, 289). Yksikkötestauksella yritetään pitämään huoli siitä, että ohjelman pienimmät osat toimivat vaatimusten mukaisesti. Tämän tyyppiset testit ovat usein automatisoituja ja ohjelmistokehittäjän kirjoittamia. Yksikkötestauksella ei voida yksistään suorittaa ohjelmiston osan toimivuutta, mutta sillä voidaan ennakoida virhetilanteita. Yksikkötestauksen hyödyt tulevat esiin erityisesti, jos koodiin joudutaan tekemään muutoksia. Sillä kun testaus on valmiiksi automatisoitu, tehdyllä yksikkötestillä virheet ovat helposti löydettävissä (Smarteducation 2017).

Integraatiotestaus on ohjelmiston testaamista komponenttien ja niiden rajapintojen välillä (kuva 11). Tämän tarkoituksena on löytää ohjelmistovirheitä, joita ei saada selville pelkällä yksikkötestauksella. Integraatiotestauksen tulokset ovat yleensä verrattavissa teknisen määrittelyn mukaisesti (Istqbexamcertification 2017).

Integroititestausta etenee tyypillisesti yhdessä yksikkötestauksen kanssa, tämän takia sitä harvoin tarkastellaan erillään yksikkötestauksesta. Normaalisti integroititestausta esitetään etenevän bottom up (suom. kokoavasti) alimman tason yksiköistä ylöspäin. Top down (suom. jäsentävästi) integroinnissa etenemissuunta on alaspäin (Haikala & Märijärvi 2004, 290).



KUVA 11. Integrointitestausta (Istqbexamcertification 2017, Mukailen)

Järjestelmätestauksessa tarkastellaan kokonaista ohjelmaa. Testauksen tavoitteena on tarkastaa, vastaako se sille annettuja vaatimuksia ja käyttötarkoitusta. On tärkeää, että järjestelmää testataan sen ”aidossa” ympäristössä, tällä tarkoitetaan sitä, että testauksen suorittajat ovat mahdollisimman riippumattomia sen kehitystyöstä.

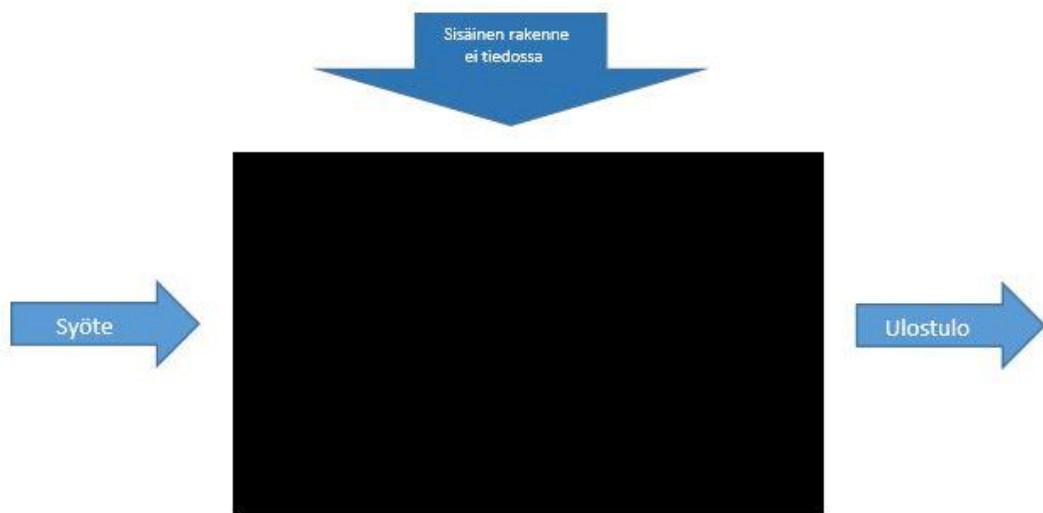
Myös toimintaan liittymättömät testaukset ovat osa järjestelmätestausta, tällaisia ovat esimerkiksi lokalisoititestausta, käytettävyydestausta, kuormitustestausta ja luotettavuustestausta. Käytettävyydestausta ja lokalisoititestausta käsitellään tarkemmin luvussa 3.4 Testaustavat.

Mitä ylempänä V-mallin testaustasoilla ollaan, niin sitä suuremmat ovat virheiden korjauksesta aiheutuvat kustannukset. Kun virheitä korjataan, niin se voi aiheuttaa ohjelmistossa uuden virheen. Esimerkiksi järjestelmätestauksen yhteydessä löydetty virhe korjataan niin se aiheuttaa muutoksia ohjelmiston muissa yksiköissä. Tämän jälkeen yksiköt on testattava uudelleen, jonka jälkeen järjestelmätestausta suoritetaan kokonaan uudelleen. Tällaista kutsutaan regressiotestaukseksi (Haikala & Märijärvi 2004, 289-290.)

3.4 Testaustavat

Ohjelmiston testauksesta käytetään usein laatikkometaforaa. Nämä testaustavat ovat black-box (musta laatikko), white-box (valkoinen laatikko) ja grey-box (harmaa laatikko). Edellä mainitut ovat yleisimmät testaustavat.

Black-box testaus on ohjelmistontestaus menetelmä, jossa ohjelman sisäinen rakenne, suunnittelu ja toteutus eivät ole testaajan tiedossa. Nämä testit voivat olla toiminnallisia tai ei-toiminnallisia, mutta useimmiten toiminnallisia. Menetelmä on saanut nimensä siitä, että ohjelmisto on testaajalle kuin musta laatikko jonka sisälle ei voi nähdä (kuva 12). Esimerkiksi testaaja testaa selaimella internetsivua, kuten hiirellä klikkaamista ja selvittää onko sen toimivuus odotetun lainen. Testaaja suorittaa tämän ilman minkäänlaista tietoa sivuston sisäisestä rakenteesta (Softwaretestingfundamentals 2017).

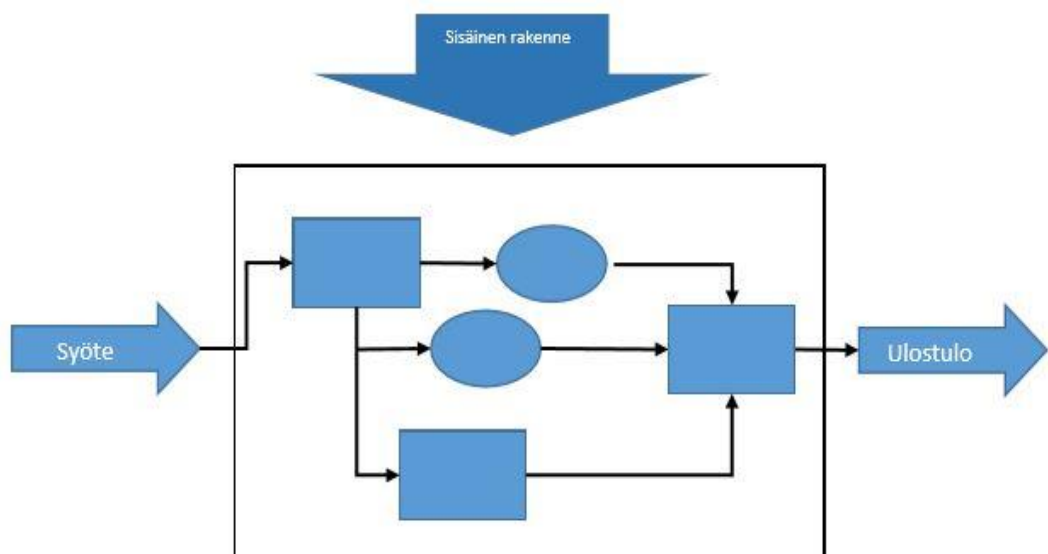


KUVA 12. Black-box testaus (Tutorialspoint 2017, Mukailten)

Tämän testausmenetelmän etuina voidaan pitää sitä, että testi tehdään ”oikean” käyttäjän näkökulmasta. Eduksi katsotaan myös, että testaajan ei tarvitse osata ohjelmointikieliä tai tietää miten ohjelmisto on toteutettu. Näin testejä voidaan suorittaa kehittä-

jistä riippumattomana, saadaan tätä kautta puolueeton näkökulma toimivuuteen (Softwaretestingfundamentals 2017).

White-box (kuva 13) on ohjelmistotestaus menetelmä, joka tunnetaan myös nimillä clear-box, open-box, glass-box testaus. Tämän opinnäytetyön teoriaisuudessa käytetään nimeä white-box testaus. Tämä menetelmä saa nimensä siitä että, testaajan silmissä laatikko ajatellaan ”läpinäkyvänä”, jonka sisälle testaaja näkee toisinkuin black-box testauksessa (Softwaretestingfundamentals 2017).



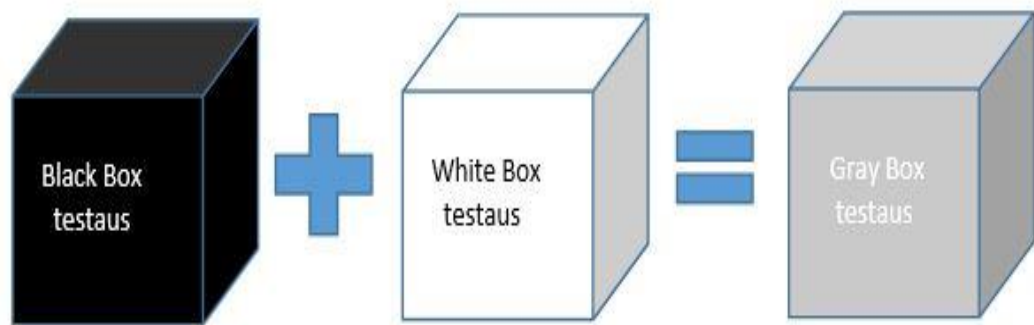
KUVA 13. White-box testaus (Tutorialspoint 2017, Mukailen)

White-box testauksessa, testattavan ohjelmiston sisäinen rakenne, suunnittelu ja toteutus ovat jo entuudestaan testaajan tiedossa. White-box testaus menetelmää sovelletaan usein yhdessä seuraavien ohjelmistotestausten tasojen kanssa näitä ovat yksikkötestaus, integraatiotestaus ja järjestelmätestaus. Pääasiassa kuitenkin yksikkötestauksessa.

Tämän menetelmän etuina voidaan pitää sitä, että testaus on mahdollista aloittaa aikaisessa vaiheessa, sillä GUI:ta (suom. graafinen käyttöliittymä, tulee sanoista graphical user interface) ei tarvita. Haitaksi voidaan katsoa testauksen vaikeus, sillä testit voivat olla hyvin monimutkaisia ja vaativat näin osaavaa henkilökuntaa, joilla on perusteelli-

nen tuntemus ohjelmoinnista ja sovelluksen kehittämisestä (Softwaretestingfundamentals 2017.)

Grey-box testausmenetelmä on yhdistelmä/sekoitus black-box ja white-box menetelmää (kuva 14). Aikaisemmissa käytiin läpi edellä mainittujen testausmenetelmien eroja. Grey-box testauksessa sisäinen rakenne on osittain tunnettu. Tässä menetelmässä edellytyksenä on siis pääsy osittain tietorakenteisiin. Usein testaus kuitenkin tapahtuu black-box tasolla. Menetelmä saa nimensä siitä, että testaajan silmissä ohjelma ajatellaan ”läpikuultavana” laatikkona jonka sisälle voi osittain nähdä (Softwaretestingfundamentals 2017).

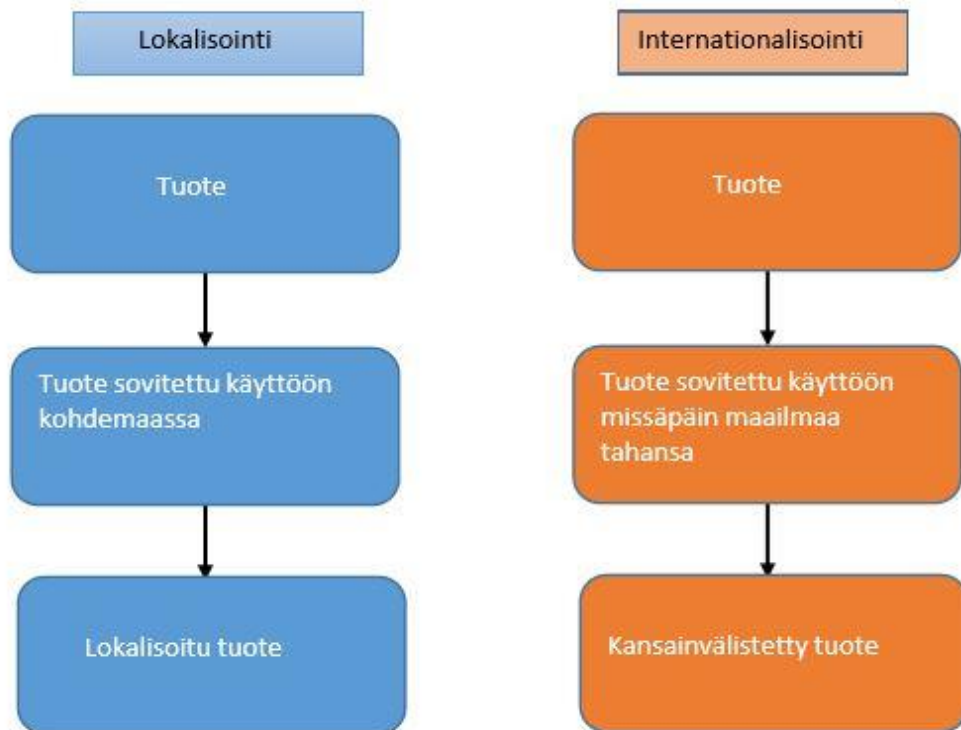


KUVA 14. Gray-box testaus (Tutorialspoint 2017, Mukaillen)

Käytettävyystestauksen (usability testing) tavoite on tarkistaa, pystyykö ohjelmistoa käyttävä henkilö suoriutumaan hyvin niistä tehtävistä, joita varten ohjelmaa on rakennettu. Pääasiassa tämäntapainen ei-toiminnallinen testausmuoto on käyttöliittymätestausta (Haikala & Märijärvi 2004, 291). Muita ei-toiminnallisia testaustapoja ovat mm. lokalisointi.

Tietojenkäsittelytieteessä puhutaan internationalisoinnista (kansainvälistäminen) ja lokalisoinnista (kotoistus). Ne ovat prosesseja joilla mukautetaan erilaisia tuotteita

vieraaseen ympäristöön, kuten ohjelmistoja. Ohjelmiston lokalisointi on yksi näkyvimmistä ja kriittisimmistä osa-alueista ohjelmiston julkaisun kannalta.



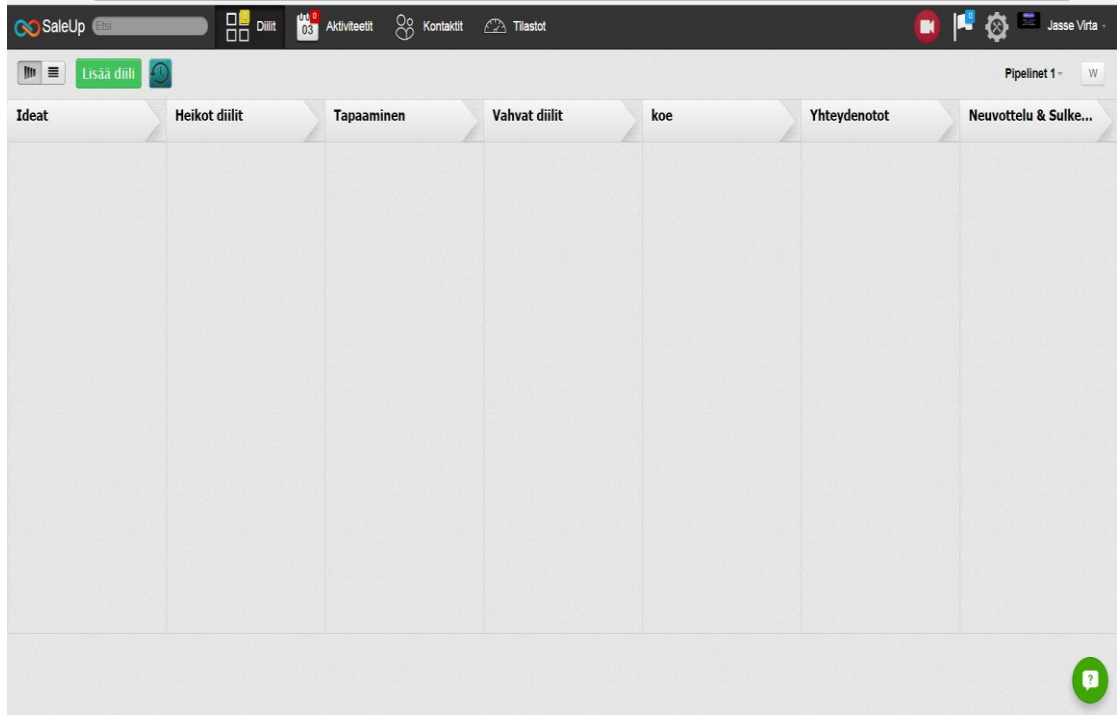
KUVA 15. Lokalisointi & Internationalisointi (Guru99 2017, Mukailen)

Internationalisoinnin ja lokalisoinnin ero on vähäinen (kuva 15). Internationalisointi on erilaisten tuotteiden sovittamista käyttöön missä päin maailmaa tahansa. Lokalisointi puolestaan tarkoittaa tuotteiden sovittamista kohdemaahan (W3 2015).

Lokalisointitestausta on osa ohjelmiston testausprosessia. Lokalisointitestausta on tarkoitus tarkistaa, kuinka hyvin ohjelma on käännetty sen kohdekielille ja kulttuuriin sopivaksi. Testausta tehdään käyttöliittymää ja sen sisältöä testaamalla. Lokalisointitestausta on, ei-toiminnallista testaamista (Guru99 2017).

4 CRM-OHJELMISTON LOKALISOINTITESTAUS

Roihu Solutions Oy on suomalainen ohjelmistoalan startup-yritys, jonka yksi tarjoamista tuotteista on Amazonin pilvipalvelussa toimiva vietnamilainen CRM-ohjelmisto nimeltään SaleUp (kuva 16). Roihu Solutions vastaa ohjelmiston lokalisoinnista ja tuomisesta Suomen markkinoille. Menestyäkseen tuotteen täytyy olla onnistuneesti sovitettu vastaamaan suomalaisten yritysten sekä asiakkaiden tarpeita.



KUVA 16. Kuvankaappaus SaleUp-ohjelmiston käyttöliittymästä

Roihu Solutionin suorittama CRM-ohjelmiston lokalisointi alkaa olla suurimmaksi osaksi valmis ja yritys on siirtymässä lokalisoinnin ja käytettävyyden testaamisen pariin. Tulevien testauksien on tarkoitus löytää mm. käänkövirheitä, suomentamattomia osioita ja mahdollisesti muita sellaisia asioita jotka ovat jääneet kehittäjältä huomaamatta. Testauksesta saatavalla palautteella Roihu Solutions voi poistaa mahdolliset virheet tuotteesta.

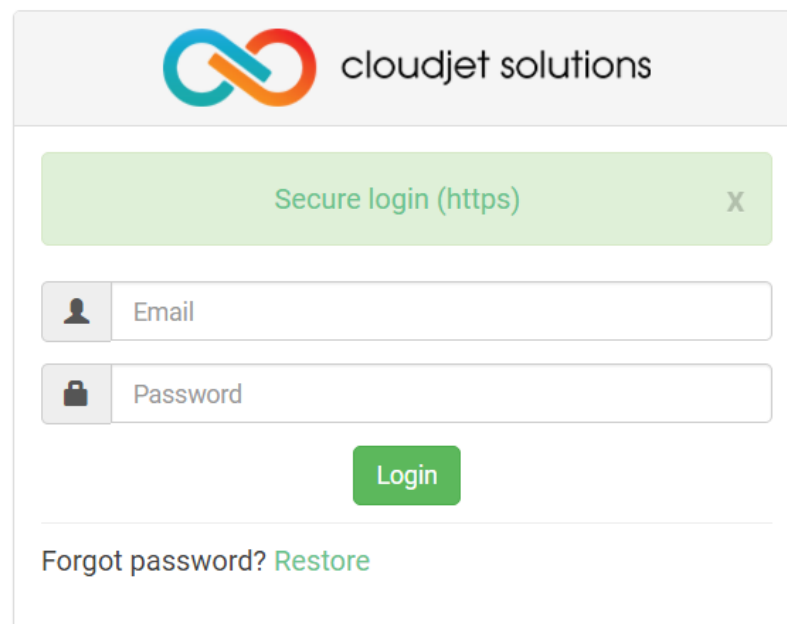
Tuote on siinä vaiheessa, että se on siirtymässä testausympäristön ulkopuolelle. Roihu Solutionin sovittaman tuotteen ensimmäisenä pilottiasiakkaana toimii Rakennusliike U. Lipsanen Oy joka on rakennusurakointiin keskittynyt yritys. SaleUp on tarkoitus liittää osaksi heidän toiminnanohjausjärjestelmäänsä (ERP).

4.1 Testaussuunnitelma

Testaus tulee tapahtumaan osana järjestelmätestausta. Testaus käsittää lokalisointitestauksen joka kuuluu ei-toiminallisen testauksen piiriin. Tämän testauksen tarkoituksena on varmistaa, että tuotteessa ei ilmene käänkövirheitä ennen pilottiasiakkaalle päätymistä.

Testaus aloitetaan tutkimalla CRM-ohjelmiston käyttöliittymän rakennetta kokonaisuudessaan ja sitä kautta purkamalla se pienempiin osioihin, jotka käydään läpi testaussuunnitelman mukaan. Testaussuunnitelma luodaan lokalisointitestauksen tarpeiden mukaan.

Testaus tehdään kirjautumalla SaleUp-palveluun kirjautumissivun kautta (kuva 17). Kirjautumisen jälkeen käyttäjä pääsee palveluun sisään, jonka jälkeen päästään palvelun päänäkymään. Päänäkymässä on mahdollista navigoida sivuston eri osioissa ja toimintoissa. Toimintoihin kuuluvat yleiset CRM-ohjelmiston piirteet.



cloudjet solutions

Secure login (https) X

Email

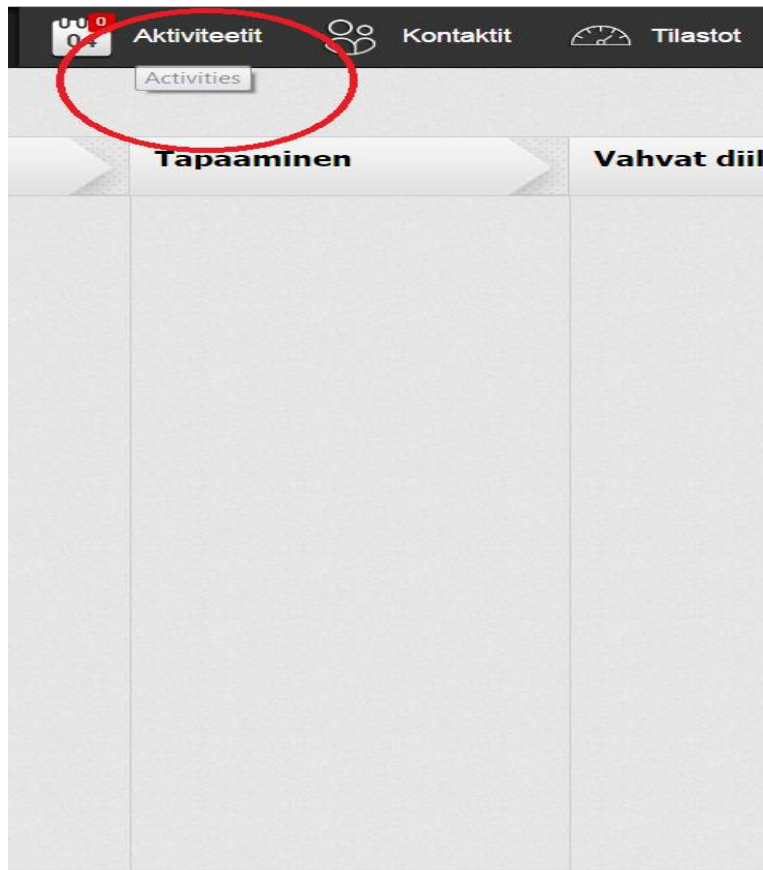
Password

Login

Forgot password? [Restore](#)

KUVA 17. Kuvankaappaus SaleUp kirjautumiskäytännöstä

Mahdollisen käännösvirheen löydyttyä siitä otetaan kuvankaappaus, josta selkeästi näkyy missä käännösvirhe sivustolla sijaitsee. Tämän jälkeen kuvamateriaaliin graafisesti merkitään tapahtunut virhe ja sen korjaamiseen tarvittava tieto. Testauksesta saatu kuvamateriaali toimitetaan Roihu Solutionin ohjelmistotestauksesta vastaavalle henkilölle. Testausvastaavan todennettua virheen todeksi, materiaali lähetetään mahdollisine liitteineen sähköpostilla Saleup-kehitystyöstä vastaaville sovelluskehittäjille.



KUVA 18. Käännösvirheen raportointi

Sovelluskehittäjät tutkivat saadun materiaalin virheestä, jonka jälkeen ohjelmistoon tehdään vaatimusten mukaiset muutokset. Virheen suuruudesta riippuen sen korjaaminen kestää 1-7 päivään. Muutostyön tehtyään kehitystiimi lähettää vahvistuksen korjauksen tekemisestä.

Roihu Solutions tarkastaa tehdyn korjauksen, jos korjaus on puutteellinen tai virheellinen pyydetään korjaus tekemään uudestaan. Kun korjaus on hyväksytty ja lisätty ohjelmistoon, se siirretään takaisin testattavaksi. Korjaus testataan ja jos se täyttää sille asetetut vaatimukset, on testaus hyväksytty. Puutteiden ilmettyä alkaa prosessi uudestaan.

4.2 Testaus

Lokalisointitestausta tehdään käymällä läpi jokainen sivu oikolukemalla tekstien suomen kielen oikeinkirjoitus, etsien suomentamattomat osiot ja muut kielelliset poikkeavuudet. Jokainen löydetty virhe taltioidaan ja merkitään testausraporttiin. Tarkoituksena on toimittaa Roihu Solutionille raportti kaikista löydetyistä virheistä mahdollisimman nopeasti ennen pilotointia.

Lokalisointitestausta aloitettiin menemällä kirjautumissivulle. Jo kirjautumissivulla oli useita testauksen läpäisemättömiä virheitä. Kirjautumissivu oli kokonaan suomeksi kääntämätön. Sivulla oli englannin ja oletettavasti myös vietnamin kielellä toimiva kirjautumissivusto (kuva 19).



KUVA 19. Kuvankaappaus virheellisestä kirjautumissivun lokalisoinnista

Sisäänkirjautumisen jälkeen huomio kiinnittyi ensimmäisenä oikeassa alareunassa olevaan vihreään ”help”- nappiin. Nappia painamalla sivuston keskiosaan aukesi dialogi-ikkuna, jossa oli ilmeisesti asiakaspalvelun yhteystiedot. Tämä osa oli kokonaan kääntämätön.



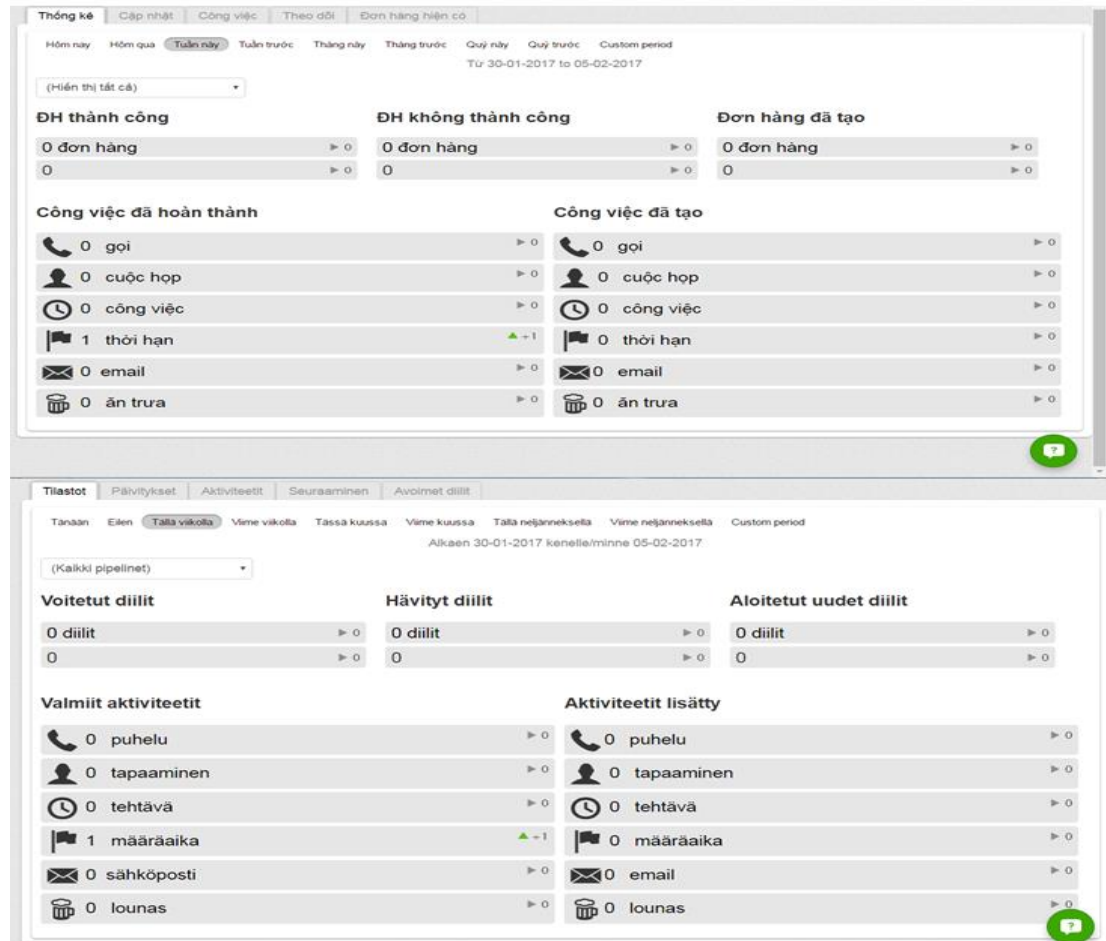
KUVA 20. Kuvankaappaus kääntämättömästä dialogi-ikkunasta.

Tämän jälkeen käytiin läpi sivuston neljä pääsivua jotka ovat diilit, aktiviteetit, kontaktit ja tilastot. Näissä sivustoissa ilmeni pieniä käänkösvirheitä ja joitakin yksittäisiä kääntämättömiä sanoja (kuva 21). Vaikka virheet olivat pieniä kokonaisuuteen nähden, eivät ne täytä ohjelmalle asetettuja vaatimuksia.



KUVA 21. Kuvankaappaus kääntövirheistä ja kääntämättömistä sanoista

Tästä eteenpäin ohjelmistossa esiintyneet virheet olivat edellisen kaltaisia pieniä käänkösvirheitä. Huolimatta virheiden pienuudesta niitä esiintyi ohjelmistossa usein kohdissa jotka saattavat häiritä sivuston käyttäjää. Kun virheet korjataan, tuloksena on lokalisoitu lopputulos (kuva 22).



KUVA 22. Kuvankaappaus lokalisoidusta lopputuloksesta

Tässä käytiin läpi testauksen aloituksessa mainitut erilaiset virhetyypit ja niiden esiintyminen Suomeen lokalisoitavassa SaleUp-ohjelmistossa. Koska virheet ovat pieniä ja tiheästi esiintyviä niiden testaaminen vaativat erityistä tarkkuutta ja huolellisuutta. Tämän huolellisen testaamisen tuloksena voidaan tarjota ohjelmiston loppukäyttäjälle luonteva tapa kokea ohjelmisto kotimaisena tuotteena sen ulkomaisesta alkuperästä huolimatta.

5 PÄÄTÄNTÖ

Opinnäytetyön tekeminen oli haastava, mutta samalla myös varsin monipuolinen ja opettava kokemus. Haasteena oli se, että koko aihealue oli minulle lähes tuntematon varsinkin CRM:n osalta. Työn tavoitteena oli saada lokalisoitua Amazonin pilvipalvelussa toimiva vietnamilainen CRM-ohjelmisto, niin että sen käyttäminen onnistuu sen kohdemaassa ohjelmiston alkuperästä huolimatta.

Ohjelmiston lokalisoititestausta vaiheessa haasteeksi muodostui sen toteuttamisen vaikeus. Toteuttaessani järjestelmätestaukseen kuuluvaa ei-toiminnallista lokalisoititestausta suureksi ongelmaksi muodostui kirjoituksen itsensä toistaminen, sillä testaus oli samanlaista käännösvirheiden etsimistä alusta loppuun. Tämän yksinkertaisen testaus-tavan takia yritin havainnollistaa testauksen mahdollisimman ei itseään toistavaksi, mutta silti ymmärrettäväksi kokonaisuudeksi.

Mielestäni opinnäytetyö oli onnistunut, vaikka ottaessani tämän työn vastaan en tien-nyt aiheesta juurikaan mitään. Työtä tehdessä opin uusia asioita niin CRM:sta kuin myös ohjelmistotestauksesta. Lokalisointitestauksesta täytynee sanoa, vaikka työ oli paikoin puuduttavaa sekä itsensä toistamista niin yritin tehdä hommani parhaani mukaan. On ilo, että sain olla mukana lokalisoititestaajana Roihu Solutionille ja luulenkin, että testauksesta saaduilla virheraportoinneilla Roihu Solution on askeleen lähempänä ja valmiimpana tuotteen pilotointia. Lopuksi tahdon vielä sanoa kiitos Pertti, thank you Perth.

LÄHTEET

Asiakkuudenhallinnan analytiikka pähkinänkuoressa.2015. Louhia. WWW- dokumentti. <http://www.louhia.fi/2015/02/19/asiakkuudenhallinnan-analytiikka-pahkinankuoressa-part-1/>. Ei päivitystietoja. Luettu 1.12.2016.

Oksanen, Tommi 2010. CRM ja muutoksen tuska. Helsinki: Talentum Media Oy.

Mäntyneva, Mikko 2001. Asiakkuudenhallinta. Vantaa: WSOY.

Haikala Ilkka, Märijärvi jukka 2004. Ohjelmistotuotanto. Helsinki: Talentum Media Oy.

Tiirikainen, Vesa 2010. IT ja parempi bisnes. Helsinki: Talentum Media

Järjestelmät. 2013. Tivi. WWW-dokumentti. <http://www.tivi.fi/CIO/2013-04-23/Aimik%C3%A4-crm-Suomessa-luotetaan-yh%C3%A4-Exceeliin-3200309.html> Päivitetty 23.4.2013. Luettu 6.12.2016.

Gartner CRM market Share Analysis. 2016. Forbes. WWW-dokumentti. <http://www.forbes.com/sites/louiscolombus/2016/05/28/2015-gartner-crm-market-share-analysis-shows-salesforce-in-the-lead-growing-faster-than-market/#6ac309c225ae> Ei päivitystietoja. Luettu 6.12.2016.

Zapps 2014. Zapps. WWW-dokumentti <http://www.zapps.fi/wp-content/uploads/Opas-onnistunut-crm-kayttoonotto.pdf>. PDF-dokumentti. Päivitetty 15.10.2014. Luettu 6.12.2016.

Kuulostaa hienolta. 2012. Myynti20. WWW-dokumentti. <http://www.myynti20.fi/asiakkuudenhallinta-crm-mista-on-kysymys/>. Päivitetty 3.9.2012. Luettu 7.12.2016.

Why Read This Report. 2009. Forrester. WWW-dokumentti. <https://www.forrester.com/report/How+To+RiskProof+Your+Culture+For+CRM/-/E-RES48381>. Päivitetty 27.2.2009). Luettu 8.12.2016.

Challenges of CRM. 2016. Google. WWW-dokumentti. <https://sites.google.com/a/siena.edu/customer-relation-management-crm-systems/challenges-of-crm>. Ei päivitystietoja. Luettu 7.12.2016.

CRM software spending by vendor.2016. Gartner. WWW-dokumentti. <http://www.gartner.com/newsroom/id/3329317>. Päivitetty 25.6.2016. Luettu 7.12.2016.

People issues. 2016. Superoffice. WWW-dokumentti. <http://www.superoffice.com/blog/crm-end-user-adoption/> Päivitetty 4.9.2016. Luettu 7.12.2016.

14 Years of CRM Failure. 2015. Ecaminc. WWW-dokumentti. <http://ecaminc.com/index.php/blog/item/439-your-crm-implementation-is-going-to-fail>. Päivitetty 24.9.2015. Luettu 9.12.2016.

Compare.2016. G2crowd. WWW-dokumentti.

<https://www.g2crowd.com/compare/microsoft-dynamics-crm-vs-salesforce-crm-vs-sugarcrm>. Ei päivitystietoja. Luettu 11.1.2017.

Compare.2016. G2crowd WWW-dokumentti.

<https://www.g2crowd.com/compare/base-crm-vs-highrise-vs-insightly-vs-pipedrive>. Ei päivitystietoja. Luettu 11.1.2017.

Screenshots.2016. Technologyadvice. WWW-

dokumentti.<http://technologyadvice.com/products/pipedrive-reviews/>. Ei päivitystietoja. Luettu 11.1.2017.

Crm-järjestelmistä.2014. Ampiiiri. WWW-dokumentti.

<https://ampiiiri.com/2011/01/07/crm-jarjestelmista-ja-sellaisen-valinnasta/>. Päivitetty 10.1.2014. Luettu 12.1.2017.

CRM-vertailu.2016. Vinkkisolution. WWW-dokumentti.

<http://vinkkisolutions.com/fi/arkistot/960#summary> Ei päivitystietoja. Luettu 12.1.2017.

Types Of CRM.2017. Techonestop. WWW-dokumentti. <http://techonestop.com/types-of-crm-operational-analytical-collaborative> Ei päivitystietoja. Luettu 12.1.2017

Ei päivitystietoja. Luettu 12.1.2017

Difference between.2017. Techonestop. WWW-dokumentti.

<http://techonestop.com/what-is-analytical-crm> Ei päivitystietoja. Luettu 13.1.2017.

Aalyttinen CRM.2006. Tivi. WWW-dokumentti. <http://www.tivi.fi/Arkisto/2006-06-07/Crm-hallitsee-asiakkuudet-3087562.html> Päivitetty 7.6.2006. Luettu 13.1.2017.

Luettu 13.1.2017.

Collaborative CRM.2017. Searchsalesforce. WWW-dokumentti

<http://searchsalesforce.techtarget.com/definition/collaborative-CRM> Ei päivitystietoja. Luettu 13.1.2017.

Data Requirements.2017. Tutorialspoint. WWW-dokumentti.

https://www.tutorialspoint.com/customer_relationship_management/crm_quick_guide.htm Ei päivitystietoja. Luettu 13.1.2017.

Strategic CRM.2017. Tutorialspoint. WWW-dokumentti.

https://www.tutorialspoint.com/customer_relationship_management/crm_types.htm. Ei päivitystietoja. Luettu 18.1.2017.

Francis, Buttle 2008. Customer Relationship Management. Oxford: Linacre House.

Methodologies of software testing.2017. Softwaretestingfundamentals. WWW-

dokumentti. <http://softwaretestingfundamentals.com/software-testing-methods/>. Ei päivitystietoja. Luettu 24.1.2017.

Ohjelmiston käyttöönotto. WWW-dokumentti.2017. Xtract.

<http://xtract.fi/kayttoonotto-tai-julkaisu-ohjelmiston-elinkaaren-vaiheena/>. Ei päivitystietoja. Luettu 24.1.2017.

Toteutus. WWW-dokumentti. 2002. Joensuu.
http://cs.joensuu.fi/~jimmonen/jot_moniste/jot_moniste_121.html. Päivitetty 10/2002.
Luettu 26.1.2017.

This day. WWW-dokumentti. <http://www.computerhistory.org/tdih/September/9/>. Ei päivitystietoja. Luettu 26.1.2017.

Forsberg Kevin, Mooz Harold 1991. The Relationship Of System Engineering To The Project Cycle. PDF-dokumentti.
<http://www.damiantgordon.com/Videos/ProgrammingAndAlgorithms/Papers/The%20Relati-onship%20of%20System%20Engineering%20to%20the%20Project%20Cycle.pdf> Ei päivitystietoja. Luettu 23.1.2017.

Testauksen tasot.2017. Smarteducation. WWW-dokumentti.
<http://smarteducation.jyu.fi/projektit/systech/Periaatteet/suunnittelun-periaatteet/testaus/testauksen-tasot>. Ei päivitystietoja. Luettu 23.1.2017.

Localization vs. Internationalization.2017. W3. WWW-dokumentti.
<https://www.w3.org/International/questions/qa-i18n>. Päivitetty 4.9.2015. Luettu 26.1.2017.

Globalization testing vs localization testing.2017. Guru99. WWW-dokumentti.
<http://www.guru99.com/globalization-vs-localization-testing.html>. Ei päivitystietoja. Luettu 2017.

What is integration testing.2017. Istqbexamcertification. WWW-dokumentti.
<http://istqbexamcertification.com/what-is-integration-testing/>. Ei päivitystietoja Luettu 31.1.2017.

What is waterfall model.2017. Testingfreak. WWW-dokumentti.
<http://testingfreak.com/waterfall-model-software-testing-advantages-disadvantages-waterfall-model/>. Ei päivitystietoja. Luettu 31.1.2017.

What is Grey box testing.2017. Tutorialspoint WWW-dokumentti.
https://www.tutorialspoint.com/software_testing_dictionary/grey_box_testing.htm. Ei päivitystietoja. luettu 31.1.2017.

Software testing overview.2017. Tutorialspoint. WWW-dokumentti.
https://www.tutorialspoint.com/software_engineering/software_testing_overview.htm. Ei päivitystietoja. Luettu 31.1.2017.

LIITE 2(1).

Monisivuinen liite