

Opinnäytetyö (AMK)

Tietojenkäsittely Tradenomi

2019

Jussi Kungas

RAPORTOINTISOVELLUKSEN LAAJENTAMINEN

– Case: Kulku www-Sovellus

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely Tradenomi

Joulukuu 2019 | 27 sivua

Jussi Kungas

RAPORTOINTISOVELLUKSEN LAAJENTAMINEN

- Case: Kulku www-Sovellus

Opinnäytetyön tavoitteena on tutkia www-sovelluksen laajentamista sovellukselle, jonka nimi on "Kulku". Kehitysprosessin aikana ilmeni erilaisia ongelmia, joita me hyödynsimme tutkiessamme uusia ratkaisuja järjestelmän rakennemallin parantamista varten. Työssä kyseenalaistettiin laajennuksen pääongelmaa.

Projektin toimeksiantajana oli Suomen It-ratkaisut Oy, suomalainen moniosaaja yritys, joka keskittyy tietotekniikkaan ja tietoteknisiin palveluihin. Työn aikana kuvattiin yrityksen toimintatapoja ja, sitä miten yrityksen vaikutus näkyi työprosessissa.

Tuloksena saimme selville, miten suunnitelma toteutui käytännössä, mitä ongelmia siinä oli ja mitä opimme informaation kulun ongelmista monen ohjelmiston välillä. Laajennuksen pääongelmaksi osoittautui standardin puute. Emme keksineet ratkaisua standardin puutteeseen, mutta pystyimme oletamaan, että mitä hyviä ja huonoja puolia standardin luomisesta syntyisi.

ASIASANAT:

ohjelmistokehitys, tietojärjestelmät, raportointi

BACHELOR'S | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor of Business Administration

December 2019 | 27 pages

Jussi Kungas

EXTENSION PROJECT OF A REPORTING APPLICATION

- Case: Kulku WWW Application

The purpose of this thesis is to investigate the creation process of the extension for a www application made by Suomen It-ratkaisut Ltd: a Finnish multi-talented company which focuses on IT and IT services.

The thesis goes through the development process of a www application extension dubbed "Kulku". The process was described step by step from planning to completion while also describing the work environment within the development team and the company. The problems were mostly project-specific thus they were iterated upon during the development.

The project was developed with a team of two people. I was the main developer and the second developer was the project owner itself. The program was left on the backburner by the project owner and I was instructed to develop a better solution for the reporting extension's system. During this development period I designed a system that was practically an untangled version of the old system thanks to the reduced amount of repetition. During this design process I started noticing how the underlying cause for the reporting system's existence might have been due to the lack of standardization of data handling within different companies. Even though the main cause for the reporting extension's existence was revealed, the research for it proved inconclusive.

The development was unorthodox thanks to the development team setup, but it did not hinder the project due to it being akin to a collaboration project in practice. During the system design we noticed that separation of concerns and reduction of repetition caused increased efficiency every time it was applied to the processes of the system. Separation of concerns made it easier for the developers to work with the project and reduction of repetition made the system more stable and modular. Even though we learned that the reason for the extension's existence was more complicated than anticipated, we currently have no real answer for it, but we believe there is something that could be changed in the future.

KEYWORDS:

software development, information systems, reporting

SISÄLTÖ

KÄYTETYT LYHENTEET	6
1 JOHDANTO	7
2 SUOMEN IT-RATKAISUT TOIMINTA-AJATUS	9
2.1 Projektin elinkaari	9
2.2 Sovelluksen tuki ja elinikä	10
3 TYÖNJAKO	11
3.1 Projektioorganisaatio	11
3.2 Projektinhallinta	11
4 SUUNNITTELU JA UUSIMINEN	13
4.1 Ongelman havaitseminen	13
4.2 Suunniteltu ratkaisu	14
5 TIETOTURVAN KEHITTÄMINEN	16
5.1 Mikä on Owasp?	16
5.2 Owasp käytännössä	16
5.3 XSS -hyökkäykset ja niiden esto	17
5.4 Virheiden käsittely	17
6 PROJEKTIN PÄÄONGELMA	19
6.1 Miksi standardisoida?	19
6.2 Standardisoinnin huonot puolet	19
6.3 Toiminta järjestelmien yhdistäjänä	20
6.4 Johtopäätös	20
7 VIIMEISTELYVAIHE	22
7.1 Näytä paljon ja vähän	22
7.2 Huolenaiheiden erottamisen vaikutus käyttöliittymään	23
7.3 Tulevaisuuden muutokset	24
8 YHTEENVETO	25
LÄHTEET	27

KUVAT

Kuva 1. Projektin kehityssilmukka.	10
Kuva 2. Vanha raportointijärjestelmä.	13
Kuva 3. Uusi raportointijärjestelmä.	14
Kuva 4: Vanha raportointityökalu.	22
Kuva 5: Uusi raportointityökalu	23

KÄYTETYT LYHENTEET

Owasp	The Open Web Application Security Project, tietoturvan toteutukseen keskittynyt avoin järjestö. (OWASP.)
XSS	Cross-site scripting. Tietoturva-aukko, millä suoritetaan haitallista koodia www-sivuilla. (OWASP.)
PHP	Hypertext Preprocessor. Palvelinpuolen ohjelmointikieli, joka mahdollistaa www-sivujen dynaamisen luomisen. (Php.net.)
CSS	Cascading Style Sheets, www-sivujen ulkoasun merkintää sisältävä tiedostotyyppi. (W3C.)
HTML	Hypertext Markup Language. Standardoitu www-sivujen kuvauskieli. (W3C.)
AJAX	Asynchronous Javascript And XML. Kokoelma ohjelmointitekniikoita joilla pystytään luomaan asynkronisia kutsuja JavaScriptillä. (W3Schools.)

1 JOHDANTO

Opinnäytetyö toteutettiin yritykselle Suomen It-ratkaisut Oy, sähköisen markkinoinnin ja internetsovellusten toteuttajalle. Olin työskennellyt yrityksessä ennen opinnäytetyön projektin alkua. Yrityksen käytäntöjen tunteminen helpotti projektin etenemistä.

Opinnäytetyön valinta perustui omien ohjelmointitaitojeni kehitykseen, kiinnostukseeni ja omien kykyjeni hyödyntämiseen. Ohjelmointi on oma kiinnostuksen kohteeni, ja sen takia olen valinnut tietotekniikan omaksi ammattipolukseksi. Itsekehityksen osuuden tarkoituksena oli kehittää omia taitoja www-sovelluksen laajentamisen kanssa. Oma kehitykseni ilmeni projektin aikana koettujen haasteiden selvittämisessä.

Opinnäytetyön löytämiseksi hyödynsin omaa taitoja verkostoitua. Työnantajani ja täten projektin omistaja oli henkilö, jonka kanssa olin työskennellyt moneen kertaan ennen opinnäytetyön alkua.

Opinnäytetyön projektina oli luoda laajennus, joka mahdollisti monipuolisten raporttien luomisen. Laajennus luotiin www-sovellukselle nimeltään "Kulku". Kulku www-sovellus on tietojenkäsittelyyn keskittynyt sovellus, joka mahdollistaa yrityksen tietojenkäsittelyn prosessoimisen internetistä käsin ilman erillistä latausta tai asennusta. Sovellus itse on pääasiallisesti keskittynyt yritysten työaikojen seuraamiseen ja automaattiseen kulunvalvonnan hallitsemiseen. Sovelluksen toimintoihin kuuluu muun muassa tapoja laskea asiakkaiden työaikoja ja seurata työtehtävien suorituksia. Kulku www-sovellus on monen käyttäjäyrityksen käytössä, joten sovelluksen on hallittava monen yrityksen tietoja samanaikaisesti. Sovellus palvelee kyseisiä asiakasyrityksiä asiakaskohtaisilla palveluilla. Asiakaskohtainen palvelumenetelmä oli projektin tavallisin haaste, koska jokaiselle asiakkaalle oli omat räätälöidyt sivut ja järjestelmät. Näitä järjestelmiä ei ollut välttämättä standardisoitu eri asiakkaiden välillä, mikä aiheutti asiakkaisiin liittyviä haasteita järjestelmän kehittämisessä. Laajennuksen toteuttaminen aloitettiin, koska raporttien luomiseen tarkoitettujen palveluiden tulee olla asiakaskohtaisia samalla tavalla kuin muut www-sovelluksen palvelut.

Opinnäytetyön projektia toteutti kahden henkilön projektiryhmä: minä ja projektin omistaja. Projekti toteutettiin käyttämällä hyvin vapaamuotoista versiota scrum projektinhallinnan viitekehyksestä. Minulle annettiin projektin alussa likimääräinen projektia kuvaava suunnitelma. Projektin edetessä tehtävät annettiin scrumin tavoin sopivan kokoisina

määrinä. Toisin kuin normaalissa scrum käytännössä, tehtäville ei annettu tiettyä aikarajaa.

Sovelluksen kehittämisen oli suoraviivaista. Projektin laajuus oli hyvin tiukasti rajattu verrattuna sovelluksen kokonaisuuteen, täten soveltamamme kehityssilmukka oli hyvin epätavallinen.

Tietoturvan toteutus seurasi The Open Web Application Security Project eli OWASP -standardeja. Owasp on avonainen, ilmainen ja maailmanlaajuinen tietoturvaan liittyvä tiedonlähde. Hyödynsin sitä, koska se on suurimmaksi osaksi puolueeton ja helposti konsultoitava. Tietoturvan toteuttaminen perustui Owasp tietoturva ohjeiden hyödyntämiseen, koska kyseinen tiedonlähde on hyvin monipuolinen ja yksityiskohtaisesti perusteltu.

Kehittäessämme sovellusta kyseenalaistimme, että miksi Kulku www-sovellus tarvitsi pitkälle kehittyneitä raportointityökalua. Selvitimme, että raportointityökalua käytettiin kommunikoimaan Kulku www-sovellusta vastaavien sovellusten kanssa. Päätelimme tämän johtuvan standardin puutteesta, mutta emme keksineet vastausta standardin puutteeseen. Loimme johtopäätöksiä, jotka viittasivat standardisoinnin kannattavuuteen.

Testaaminen suoritettiin kokonaisuuden valmistuessa eli silloin, kun laaja järjestelmän toiminto saatiin valmiiksi. Testaaminen sisälsi standardoidut tietoturvatestetit ja tahallista ohjelman rikkomista. Loput virheet löydettiin asiakkaan palautteen avulla. Kun tietoturva, muu testaus ja laadun tarkistus projektin omistajan kanssa oli valmis, uusin versio siirrettiin testipalvelimelta sovelluksen omalle palvelimelle asiakkaiden käyttöön.

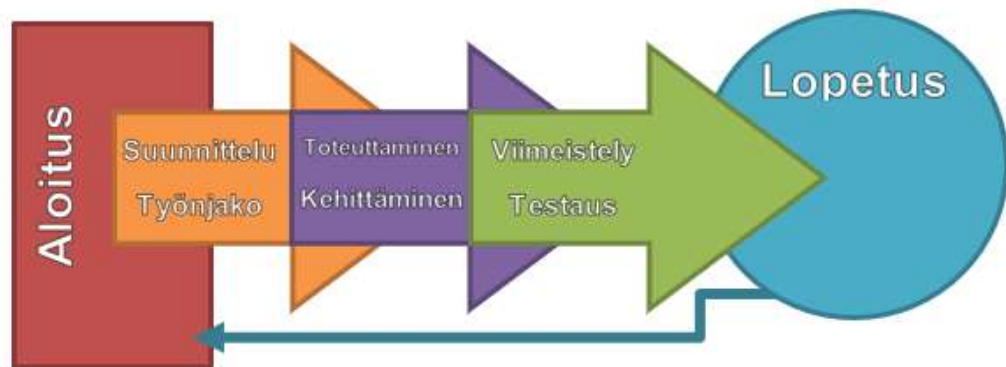
2 SUOMEN IT-RATKAISUT TOIMINTA-AJATUS

Suomen It-ratkaisut Oy on innovatiivinen kumppani sähköisen markkinoinnin ja internet-sovellusten toteuttajana. Yrityksen toimenkuvaan kuuluu PHP, CSS, MySQL ja (x)HTML-toteutus. Tarkemmin selitettynä yritys tuottaa www-ohjelmistoja ja palveluita yrityksille asiakaskohtaisella menettelyllä. Tämä ei tarkoita, että jokainen tuote tai palvelu on asiakaskohtainen tai asiakaslähtöinen. Suurin osa palveluiden tuotannosta alkaa asiakkaan pyynnöstä. (Suomen It-ratkaisut 2019.)

Suomen It-ratkaisut Oy:n projektien toiminnat ovat vakituisesti asiakaskohtaisia, paitsi silloin kun projekti ei ole asiakaslähtöinen. Näissä poikkeuksissa kyseinen hypoteettinen projekti on aloitettu potentiaalisen marketin saavuttamisen ansiosta. Tämänkaltainen poikkeus voi syntyä syistä, mutta ei rajoitettuna: markettipuutteen tunnistamiseen, laskelmoidun päätelmän toimintaan ottoon tai markkinaraon haltuun ottamiseen.

2.1 Projektin elinkaari

Projektin kehityssilmukka (Kuva 1.) alkaa tapaamisella, missä työntekijät ja työnantaja sopivat silmukan työtehtävät ja suunnittelevat, että miten kyseisiin työtehtäviin suhtaudutaan. Jälkeenpäin alkaa toteuttamisen prosessi. Toteuttamisen aikana toteutetaan sovitut työtehtävät. Toteuttamisen aikana tapahtuu myös kokeilu ja kehittäminen samanaikaisesti. Kehittämisen aikana innovaatio ja kekseliäisyys voi tapahtua luontevasti vastaantulevia ongelmia selvitettäessä. Innovaatiota on vaikeaa pakonomaisesti saada syntymään, joten sen annetaan tapahtua kehityksen aikana, kun ongelmia selvitetään. Viimeistely ja testaus olivat toistensa kanssa yhtenäinen prosessi. Testailulla saatiin systeemissä olevat virheet ilmenemään ja viimeistelyn aikana ne korjattiin. Viimeinen askel on silmukan lopetus, missä tapahtuu seuraavan askeleen päätös. Tämä päätös voi tarkoittaa, että silmukka alkaa uudelleen eri valikoimalla tehtäviä, mitkä suoritetaan seuraavassa silmukassa. Toisena vaihtoehtona on lopetus. Lopetus tarkoittaa projektin loppua, kun silmukan tehtävät ovat valmiita ja tehtäviä ei ole enää jäljellä projektia varten.



Kuva 1. Projektin kehityssilmukka.

2.2 Sovelluksen tuki ja elinikä

Suomen It-ratkaisut Oy:n toimintatapa pitkäkestoisia palveluita varten on luoda palveluita, joita voidaan laajentaa tarpeen tullen tulevaisuudessa. Palvelut suunnitellaan ja toteutetaan pitkäkestoisiksi siinä mielessä, että niiden järjestelmät ovat suunniteltu laajennettavaksi pitkäkestoisien tulevaisuuden varalta. Tämä ei välttämättä tapahdu jokaisen palvelun kanssa, mutta toteutus tehdään samalla tavalla jokaisen projektin kanssa sillä ehdolla, että palvelun elinikä ei ole erillisesti rajattu hyvin lyhyeksi ennestään. Lyhytikäiset palvelut ovat yleensä rajatapauksia, joten niitä ei oteta yleisellä tasolla huomioon yrityksen toiminnassa.

Pitkäikäisten sovellusten tuki sisältää virheiden korjauksia, järjestelmäpäivityksiä ja toiminnallisia lisäyksiä tarpeen mukaan. Palveluina ne ovat suunniteltu olemaan täysin toiminnallisia ensimmäisestä päivästä lähtien.

3 TYÖNJAKO

Suoritin projektia etätöinä, joten kokemukseni yrityksen työympäristöstä on rajallinen. Yrityksen toiminta oli hyvin erilaista kuin mikä olisi odotettavissa isommalta yritykseltä. Tarkin tapa kuvata kokemusta olisi sanoa, että kaikki toiminnot vaikuttivat yksinkertaisemmalta kuin mitä suuremmalta yritykseltä odottaisi. Tämä oletettavasti johtui projektin jäsenten määrästä. Teknisesti kaksi henkilöä työskenteli projektin kanssa, mutta suurimmaksi osaksi raportointityökalun laajentamisen toteutus kuului minulle, sillä toinen projektiryhmän jäsen oli myös projektin omistaja.

3.1 Projektioorganisaatio

Projektin omistaja ja täten myös yrityksen omistaja Marko Kairinen oli projektin omistaja eli projektin järjestäjä. Palvelun asiakkaat olivat osittain hallinnassa projektin suunnasta, mutta projektin jäsenenä en joutunut olemaan tekemisissä heidän kanssaan. Asiakkaiden toiveet välitettiin projektin omistajan kautta minulle. Auktoriteetin suuri ero työntekijöiden välillä ei vaikuttanut projektin tuotantoa negatiivisella tavalla, koska pohjimmiltaan minun tehtäväni oli ideoida ja kehittää parempi vaihtoehto raportointityökalun toiminnan järjestykselle. Projektiryhmä oli tasa-arvoinen. Tasa-arvoisuus lisäsi motivaatiota työntekijöiden kesken. Pääasiassa kaikki ideat piti kulkea projektin omistajan läpi ennen kehitykseen siirtymistä, mutta kaikilla osapuolilla oli valta vaikuttaa kehityksen suuntaan omilla mielipiteillä ja ideoillaan.

3.2 Projektinhallinta

Käytimme scrum viitekehystä projektin toteutuksessa. Käyttämämme scrummin soveltaminen erosi parilla tavalla toiminnastaan verrattuna standardi scrum viitekehukseen. Käyttämillämme sprinteillä ei ollut tiettyä asetettua aikarajaa, esimerkiksi viikkoa tai kuukautta. Sprinttien pituus riippui sprintin sisällön kontekstista ja palaverien pitäjien kyvystä osallistua kokouksiin. Sprinttien pituus muotoutui luontevasti palaverien oletettujen päivien mukaan. Projektin omistaja ei ollut aina vastaanotettavissa, joten pidimme pääpalavereja vaihtelevasti noin parin tai kolmen viikon välein. Nämä pääpalaverit toimivat periaatteessa sprinttikatselmuksina. Päivittäistä scrumia ei pidetty, koska projektiryhmän

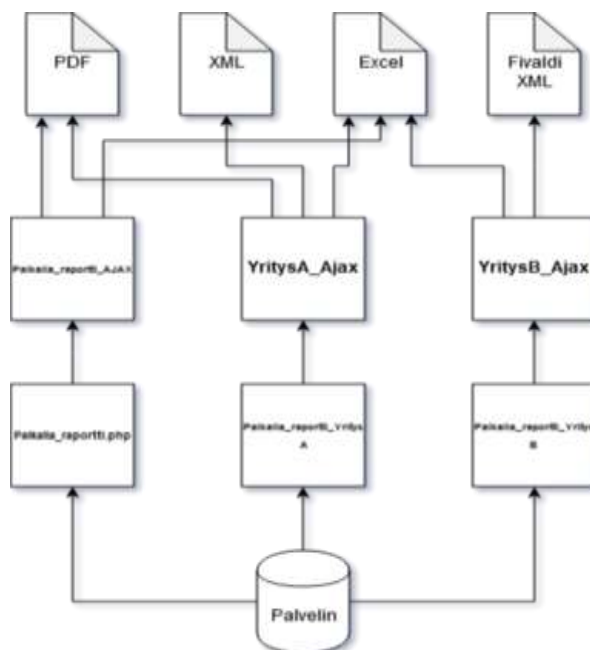
koko oli toiminnallisesti yksi. Scrummin käyttö muuten noudatti oletettua scrum toimintaa eli sprintin suunnittelua, sprintin tarkastusta ja sprint retrospektiiviä. Scrum valittiin projektin kehikseksi hyvien kokemusten vuoksi ja sen hyödyllisyyden iteraatiopohjaisissa projekteissa. (Schwaber & Sutherland 2017, 4.)

4 SUUNNITTELU JA UUSIMINEN

Tavoite oli selvä. Tarkoituksena oli suunnitella kokonaan uusi järjestelmäratkaisu hallitsemaan raporttien luontia, koska vanhassa järjestelmässä oli liian monta osaa, jotka olivat vääristyneet ajan myötä. Näiden osien toiminnalla saattoi olla vakavia seurauksia virheen sattuessa. Edellinen raportointityökalun tapa oli toimiva, mutta se ei ollut tehokas. Varsinkin jos jotain meni pieleen. Projektin aloitus oli niin sanottu paluu piirtopöydälle.

4.1 Ongelman havaitseminen

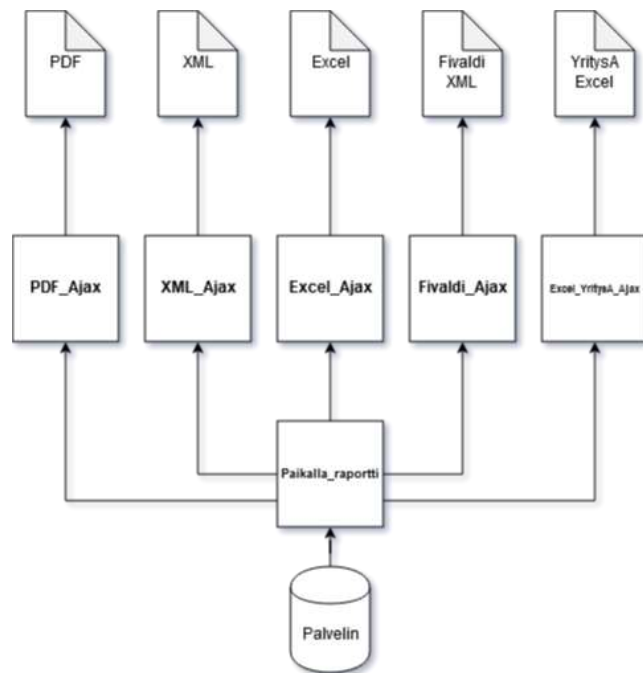
Alkuperäistä raportointijärjestelmää (Kuva 2.) tutkiessa huomasimme, että sen laajentaminen olisi kuvaannollisesti sanoen painajainen. Ongelma oli se, että jokainen yritys omasi omat sivut omien raporttien luontia varten. Nämä sivut omasivat omat ajax -sivut, mitkä tietyissä tapauksissa loivat samoja raportteja eri paikoissa. Tämä tarkoitti, että yhden raporttityypin päivittäminen tarkoitti muiden yritysten raporttisivujen päivittämistä. Kyseinen järjestelmärakenne aiheutti ylimääräistä työtä kehittäjille, siksi tämä rakenne oli muuttava kokonaisuudessaan paremmaksi jatkuvaa lisäystä ja päivittämistä varten.



Kuva 2. Vanha raportointijärjestelmä.

4.2 Suunniteltu ratkaisu

Päädyin lopullisesti keskitettyyn suunnitelmaan (Kuva3.), jossa kaikki asiakkaat loivat raporttinsa samalla sivulla eroteltuna omilla istuinnoillaan tietokannasta haetulla asiakastunnisteella. Tämä uusi rakenne mahdollisti johdonmukaisuuden eri raporttien välillä. Tätä uutta rakennetta varten tietokantaan piti tehdä muutoksia. Tietokantaan piti tehdä uusi taulu, joka hallitsi raportti -sivujen näyttämistä oikeille käyttäjille. Tällä tavalla yhdellä sivulla näkyvä raporttityyppi-lista pystyi dynaamisesti hakemaan saatavilla olevat raportointimuodot oikeille käyttäjille.



Kuva 3. Uusi raportointijärjestelmä.

Järjestelmän modulaarisointi oli oikea tapa korjata edellisen järjestelmän ongelmat. Huolenaiheiden erottaminen mahdollisti raporttityyppien lisäämisen järjestelmällisellä tasolla aiheuttamatta tiedon ylikuormitusta kehittäjille. Tämä myös mahdollisti raporttityyppien toiminnallisuuden yksilöllisyyden. Uudessa järjestelmässä yksittäisen raporttityypin toimimattomuus ei häiritse muiden raporttityyppien luomista. Toiminnan ylläpidon lisäksi, uusi järjestelmä vähensi toistoa. Toiston vähentäminen oli hyvin tärkeä muutos

tehokkuuden edistämiseksi, koska vanha järjestelmä tarvitsi muutoksen jokaista toistoa kohtaan, jos muutoksia tarvittiin yhtäkään raporttityyppiä kohtaan. (Dijkstra 1976, 1-2.)

5 TIETOTURVAN KEHITTÄMINEN

Tietoturva on tärkeä osa palvelun ohjelmointia ja tietenkin pidimme tietoturvaa tärkeänä osana ohjelmointi tuotannon kokonaisuutta. Tärkeystä huolimatta, projektin aikana tietoturvan kanssa työskentely oli hyvin vähäistä, kiitos valmiiksi asetettujen suojausmenetelmien ansiosta. Uudet projektin aikana lisätyt tietoturvamenetelmät seurasivat OWASP (The Open Web Application Security Project) menetelmiä, koska owasp on voittoa tavoittelematon, puolueeton ja maailmanlaajuinen organisaatio. (OWASP.)

5.1 Mikä on Owasp?

Owasp koostuu monen asiantuntijan ja ammattilaisen yhteistyöstä luoda puolueeton kokonaisuus tietoturvan parantamista varten. Järjestön tarkoitus on luoda ilmainen ja avoin lähde tietoturvan kehittämistä ja tutkimista varten. Owasp on monen järjestön mielestä luotettava, mukaan lukien monen opintokeskusten ja valtioiden tietoturvakeskusten mielestä. Owasp tarjoaa yhteisönsä tarjoamia ohjeita, ohjelmistoa ja tutkimuksia tietoturvan parantamista ja testaamista varten. (OWASP.)

5.2 Owasp käytännössä

Käytin owaspia tukevana ohjeena ja listana tietoturvan toteutusta varten. Minä käytin sitä suurimmaksi osaksi luomaan tarvittavat tietoturvajärjestelyt, mitkä olivat minun vastuullani. Vähäinen käyttötarve johtui käsittelemästämme datasta ja valmiiksi asetetuista tietoturva toimenpiteistä. Projektin aikana käsittelin valmiiksi luotua dataa ja uuden järjestelmän prosessit eivät luoneet tai tallentaneet uutta dataa tietokantaan. Tästä syystä raportin luomisessa ei tehty muuta kuin tarkastettu tietokannasta tuotu data varotoimenpiteiden avulla. Varotoimenpiteet suurimmaksi osaksi koostuivat tuodun datan puhdistamisesta ja virhekäsittelystä. Datan puhdistaminen suoritettiin estääkseen XSS hyökkäyksiä ja virheellisen datan korjaamista varten.

5.3 XSS -hyökkäykset ja niiden esto

XSS eli cross-site scripting on WWW-sovelluksia koskeva uhka millä suoritetaan haitallista koodia syöttämällä sitä sivulle tavalla, joka saa sivuston lukemaan koodin sivuston omilla oikeuksilla. Tämänkaltainen hyökkäys voi etenkin olla vaarallinen, jos tätä haitallista koodia tallennetaan sivuston palvelimelle. Palvelimelle tallennettua haitallista koodia pystyy ketä tahansa käyttäjä näkemään olettaen, että tallennettu koodi myös tulostetaan jollain tavalla. Tehtäviini ei kuulunut estää haitallisen koodin syöttämistä, koska en ollut vastuussa järjestelmään syötetystä datasta. Datan syöttöä ei tapahtunut raportointityökalun uusimisen aikana. Ainoa vastuualueeni oli valmiiksi tallennetun datan käsittely. XSS -hyökkäyksien estossa käytin positiivista turvallisuusmallia eli päästin vain sallittua dataa läpi. Esimerkiksi numeroita vain silloin kun odotettu tuloste oli numero. Sallimme vain tiettyjä merkkijonon merkkejä silloin kun merkkijonojen kanssa työskenneltiin. Positiivinen turvallisuusmalli valittiin, koska sen avulla pystyin suojaamaan palvelun tunnetuilta hyökkäysvektoreilta. Toinen syy positiivisen turvallisuusmallin valintaan oli negatiivisen turvallisuusmallin hallitsemattomuus. (OWASP, Manico ym 2019.)

5.4 Virheiden käsittely

Virheiden käsittely tarkoittaa virhetilanteiden hallitsemista ja vähentämistä. Virheilmoituksia voidaan käyttää hyväkseen keräämään tietoa järjestelmän teknisestä toiminnasta, joten ne ovat vaarallisia väärin hallittuina. Teknisen toiminnan tietäminen on ongelmallista, koska sitä voidaan käyttää tunkeutumaan järjestelmään. Esimerkiksi hyökkääjä voi lukea virheilmoitusten kautta rakenteen muotoa tai muuttujien nimiä. Hyökkääjän mahdollisuus kerätä tietoa järjestelmän toiminnasta voi mahdollistaa tietoturva aukon paikantamisen ja sen hyödyntämisen, joten virheilmoitusten korvaaminen mitäänsanomattomilla virheilmoituksilla on suositeltavaa. Käsittelimme virhetilanteet geneerisellä virheilmoituksella. Raportointityökalun pienen skaalan takia näytimme vain tekstin, joka sanoi: ”jokin meni pieleen”. Tällä tavalla estimme tiedon hyökkääjiltä, koska he eivät voi päätellä mitään sen kaltaisesta mitäänsanomattomasta viestistä. (OWASP, Manico ym 2019.)

Muut tietoturvaan liittyvät seikat eivät olleet minun vastuullani. Suurin osa tietoturvaan liittyvistä toimenpiteistä ja vastauksista oli projektin omistajan toteuttamia. Hyödynsin paljon valmiiksi luotuja metodeja yhdistämään uuden raportointijärjestelmän muuhun

Kulku palveluun. Kyseisissä metodeissa oli tietoturvaratkaisut valmiiksi asetettuja ja täten en pysty kuvailemaan niitä.

6 PROJEKTIN PÄÄONGELMA

Projektin aikana huomasimme, että koko kehitys keskittyi yhden ongelman korjaamisen ympärille. Raportointityökalua suurimmaksi osaksi käytettiin kommunikoimaan eri ohjelmien välillä. Kulku www-sovellus oli käytännössä viestinviejä eri asiakkaiden ohjelmistokokoelmien välillä. Kyseessä oli standardin puute ohjelmien kommunikaation kanssa, vaikka kyseiset ohjelmat käyttivät tiedostomuotoja, jotka olivat standardoituja. Ne eivät kyenneet kommunikoimaan toistensa kanssa, vaikka kaikki ulkoiset ohjelmat käyttivät samaa dataa ja samoja tiedostomuotoja. Ohjelmat lukivat kyseisen tiedon eri tavalla eri kehittäjien kehitystulosten ja menetelmien takia.

6.1 Miksi standardisoida?

Standardien olemassaolo perustuu turvallisuuteen, terveyteen ja tehokkuuteen. Standardi on todistettu aloituspiste kehitystä ja innovaatiota varten, joka toimii laajana tasoittajana itsenäisten tahojen välillä. Vain yhteisymmärryksen avulla standardisointi saadaan käytäntöön, koska standardin käyttöönotto on edellä mainittujen itsenäisten tahojen vapaaehtoinen valinta. Testaaminen muuttuu helpommaksi kun yhteiset säännöt sovitetaan standardien avulla, koska standardi on tarkoituksellisesti teknisten asiantuntijoiden konsensus. Tämä konsensus asettaa konseptille tavoitteen, jonka täyttää ennen kehityksen alkua. Kehitystyössä standardisointi mahdollistaa nykyisen kehitystyön päivityksen ja ulkoisen asiantuntemuksen konsultoinnin täten edistäen kehitystyötä monella eri tasolla. Lopullisesti standardisointi mahdollistaa uusien taloudellisten mallien kehitystä luomalla puitteet uusien innovaatioiden hyödynnettäväksi. Kuluttajat hyötyvät standardisaatiosta mahdollisuudella käyttää erilaisia tuotteita keskenään. Kuluttajien ei myöskään täydy huolehtia tuotteiden tai palveluiden laadusta, koska standardisaatio määritelmänsä mukaan luo laadun kynnyksen, jota yritykset noudattavat. (Copras project 2007, 5-6.)

6.2 Standardisoinnin huonot puolet

Standardisointi on vastaus usein toistuviin ongelmiin, joten standardin luominen luo maailmanlaajuisen vastauksen mihin tahansa ongelmaan. Tällä tavoin standardi voi evätä kaupallistetun vastauksen mihin tahansa ongelmaan. Kaupallisen vastauksen

epääminen on tarkoitettu kannustavan innovaatiota ja pakottaa kehitystä, mutta pienemät yritykset, jotka riippuivat standardin puutteesta eivät saata selvitä tämänkaltaisen radikaalin muutoksen ansiosta. Valittu standardi ei välttämättä ole paras jokaista kuluttajaa kohtaan, kyseinen valinta voi aiheuttaa tilanteen, missä tietynlainen kuluttaja ei kykene valitsemaan tuotetta tai palvelua täyttämään heidän tarpeitaan. (Shapiro & Varian 2003, 22-26.)

6.3 Toiminta järjestelmien yhdistäjänä

Ymmärsimme, että Kulku www-sovellus toimii mahdollisesti yhdistäjänä eri sovellusten kanssa. Tämänkaltaisen toiminta ei ole www-sovelluksen päätavoite, mutta kyseiselle toiminnalle oli tarve asiakkaiden palautteen mukaan. Kulku www-sovellus on asiakkaiden käyttämä pääohjelma, mutta jotkin asiakkaat käyttivät muita samankaltaisia toiminnanohjausjärjestelmiä rinnakkain ja siten tarvitsivat kyvyn käyttää näitä ohjelmia kommunikoimaan keskenään. Tällä hetkellä näiden ohjelmien kommunikaatiota ei ole standardisoitu, joten sillä on rahallista arvoa tuottaa vastaus tämänkaltaiseen ongelmaan. Raportointiin keskittynyt työkalu pystyi täyttämään tämän tarpeen, mutta siihen keskittyvä työmäärä kasvaa kasautuvasti jokaisen eri asiakkaan tarpeen mukaan. Kasvun työmäärä on riippuvainen toistosta, joten jokaisen asiakkaan tarve ei välttämättä lisää uusia raportointimalleja, koska tiettyjen asiakkaiden tarpeet menevät päällekkäin. Nykyisessä tilanteessa uusi standardin lisäys olisi haitallista Kulku -sovellusta kohtaan, koska luotu raportointi laajennus perustuu standardin puutteeseen.

6.4 Johtopäätös

Lopullisesti ongelman vastaus on tulokseton. Jokaisen ohjelman tiedostomuotojen oma-laatusuus on tapa rajoittaa kyseisten ohjelmien datan käyttöä muissa ohjelmissa. Vaikka standardisointi olisi hyödyllistä tuotannon aikana, standardin omaksuminen nykyisessä tilanteessa on kehittäjien vapaa valinta. Tämä tekee standardisoinnista vaikean ja aikaa vievän prosessin, sillä se olettaa yrityksiä yksimielisen päätöksen aloittaakseen kyseisen prosessin. Standardoinnilla on hyvät puolensa, jotka selvästi auttavat yrityksiä sekä asiakkaita. Esimerkiksi: asiakkaat hyötyisivät ohjelmien yhteensopivuudesta ja tietoturvasta. Varsinkin kun moni yritys työskentelisi saman standardin kanssa ja kehittäisivät saman tiedostomuodon tietoturvaa. Yritykset voisivat keskittää huomionsa muiden

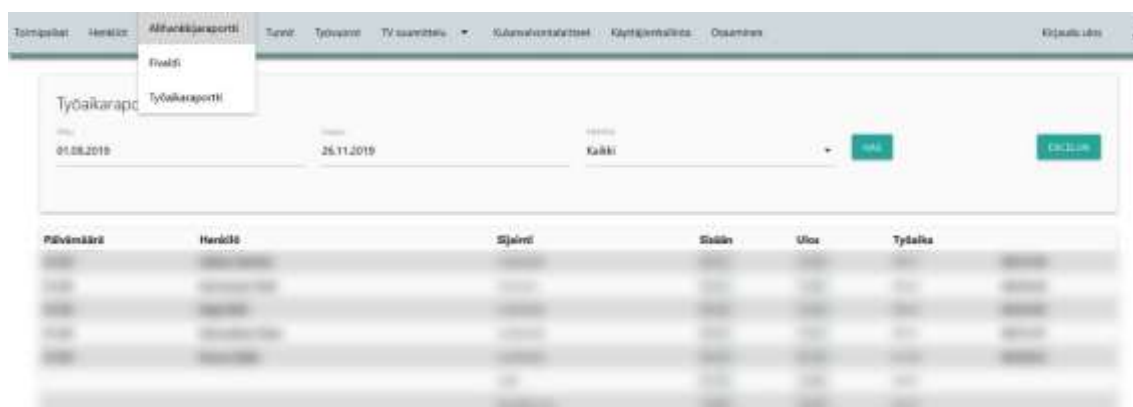
toimintojen kehittämiseen, koska tarvittava työvoima, jota olisi käytetty standardin korvaaman toiminnon kehittämiseen olisi voitu käyttää muualla. Hypoteettisessa tilanteessa standardin kehittämistä varten tarvittava työvoima olisi jaettu monen eri tahojen välillä. Nykyisessä tilanteessa standardin luominen ja käyttöönotto olisi rahallisessa mielessä haitallista Kulku www-sovellukselle, koska projektiin käytetty työmäärä olisi tehty turhaan. Joka tapauksessa: itse uskon, että kyseiseen ongelmaan olisi mahdollisuus kehittää parempi ratkaisu, kuin minkä nykyinen tilanne sallii. Paremman ratkaisun kehittäminen vaatii laajemman skaalan kehitystä ja tutkimista.

7 VIIMEISTELYVAIHE

Viimeistely alkoi, kun suunnittelu ja toteutus oli suurimmaksi osaksi saatu valmiiksi. Tämä askel prosessia keskittyi raportointityökalun käyttökokemuksen parantamiseen ja ulkoasun suunnitteluun. Käyttäjäkokemuksen kehitys perustui alkuperäisen käyttäjäkokemuksen hiomiseen poistamalla ylimääräisiä klikkauksia hyödyntämällä asiakkaan sesiota hyväkseen.

7.1 Näytä paljon ja vähän

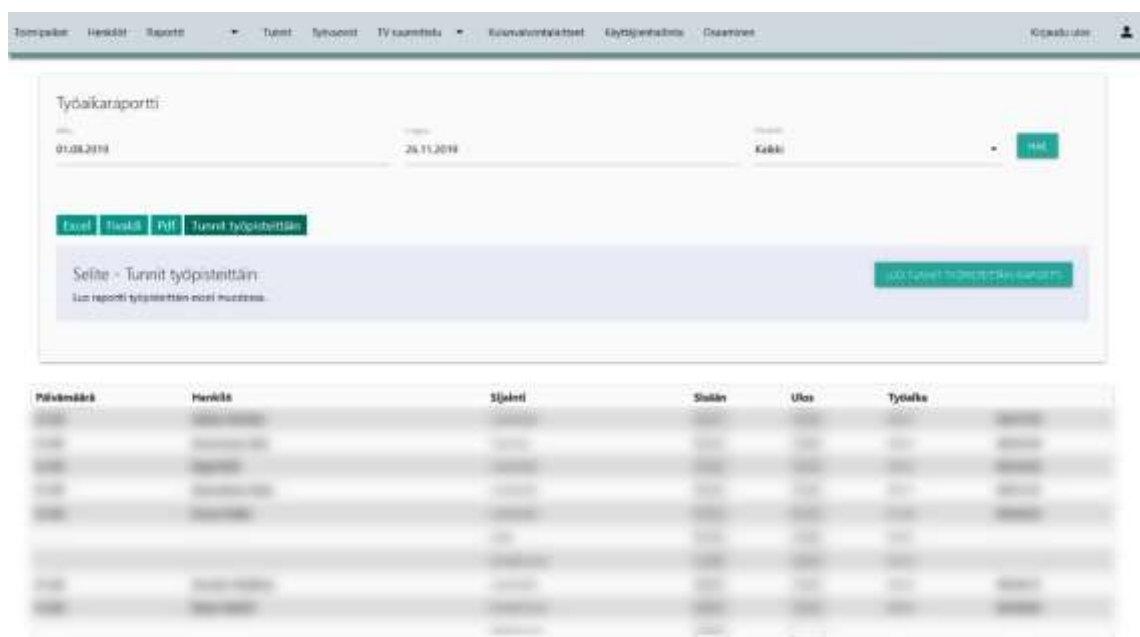
Vanhassa ulkoasussa (Kuva 4.) oli tärkeää, että kaikki tieto näytettiin siististi jaettuna omiin sivuihinsa. Tämä oli toiminnallinen tapa tehdä tämänkaltainen järjestelmä, mutta se sisälsi tiettyjä avainongelmia, mitkä olisivat haitanneet työkalun laajentamista pitkällä aikavälillä. Avattavan listan laajentaminen olisi toiminut kehityksen jatkossakin, mutta se olisi ollut hyvin riippuvainen käyttäjän laitteen pystysuorasta resoluutiosta, joten pidempi lista olisi peittänyt tärkeää dataa. Raporttien tyyppilista oli sivutuote edellisestä järjestelmästä (Kuva 2.), missä jokaisen raporttityypin luomiseen käytettiin omaa sivua. Uudessa versiossa tämän vanhanaikaisen järjestelmän oli aika siirtyä yhden sivun menetelmään (Kuva 1.). Sivun ulkoasu muutettiin kuvastamaan tätä järjestelmän muutosta.



Kuva 4: Vanha raportointityökalu.

Uuden ulkoasun (Kuva 5.) luominen aloitettiin intuitiivisesti tavoitteella, missä kaikki raportointiin liittyvät toiminnot ja tiedot löytyisi yhdestä paikasta. Halusimme, että uusi ulkoasu olisi tuttu edellisille ohjelman käyttäjille, joten emme halunneet täysin muuttaa ulkoasua. Päädyimme niin sanottuun kokoelmamalliin, missä raporttityypit oli listattu

rinnakkain. Rinnakkain listaaminen mahdollisti loputtoman raporttityyppien määrän kasvun ilman näköä estävää listaa. En halunnut turhauttaa käyttäjää ylivoimaisella määrällä informaatiota, joten päätin piilottaa raporttityyppien selitteen raporttityypin valinnan taakse, koska se on yksi pakollisista vaiheista raportin luontia varten. Pystyimme myös piilottamaan raportin luontinappulan selitteen vierelle säästääksemme kallisarvoista tilaa. Raporttityypin selite jätettiin suppeaksi ja helposti ymmärrettäväksi, jotta mahdollisemman moni käyttäjä ymmärtäisi raportin toiminnan mahdollisemman lyhyellä lukemisella. Ylimääräinen tieto voisi olla mahdollisesti hämmentävää. Ilmeisistä syistä sellaista tilannetta kannatti välttää. (Nielsen J. 1994.)



Kuva 5: Uusi raportointityökalu

7.2 Huolenaiheiden erottamisen vaikutus käyttöliittymään

Erottelun aikaansaama modulaarisuus toimi mainiosti yhden sivun menetelmän kanssa. Raportointityyppejä koskevat nappulat ja tekstit luotiin dynaamisesti tietokannasta tuodun datan avulla. Dynaaminen luonti mahdollisti käyttäjän istunnon tunnuksen hyödyntämisen käyttäjille kuuluvan datan jakamisessa. Vanha järjestelmä käytti käyttäjän istuntoa näyttämään sivuja oikeille käyttäjille, mutta uusi järjestelmä käytti istuntoa yksinomaan näyttämään oikeaa dataa käyttäjille. Data tässä tilanteessa oli dynaamisesti luodut nappulat ja niihin liittyvä tieto. Tämä tietokannan kautta luotu järjestelmä oli kehittäjäystävällisempi tapa luoda uusia raporttityyppejä, koska vanhassa järjestelmässä uuden

raporttityypin luomiseksi piti tehdä kaksi erillistä sivua: käyttöliittymän sisältävän sivun ja ajax sivun. Uudessa järjestelmässä käyttöliittymä löytyy samalta sivulta, joten ainoa uusi sivun lisäys oli ajax, joka oli linkitetty dynaamisesti tietokannan avulla nappulaan.

7.3 Tulevaisuuden muutokset

Kun suurin osa halutusta toiminnasta oli saatu valmiiksi, päädyimme keskustelemaan tulevaisuuden toiminnasta raportointityökalua kohtaan. Projektin loppupuolella huomasimme tarpeen valita alaversiota eri raporteille. Tämä johtui eri asiakkaiden tarpeista. Esimerkiksi: jotkin asiakasyritykset tarvitsivat mahdollisuuden lisätä palkkamuuttujia tiettyihin raporttityyppeihin. Kyseessä oli tapa mahdollistaa käyttäjien omien raporttien räätälöiminen. Tämä olisi ollut laaja ja aikaa vievä muutos uudelle järjestelmälle, joten sitä ei saatu valmiiksi ennen opinnäytetyön esitystä. Pidemmällä aikavälillä tämä toiminto olisi ollut mahdollista toteuttaa järjestelmän uusinnan aikana.

8 YHTEENVETO

Opinnäytetyö suoritettiin yritykselle Suomen It-ratkaisut, suomalainen sähköisen markkinoinnin ja internetsovellusten toteuttajana. Projektissa keskusteltiin Suomen It-ratkaisut Oy:n yritystoiminnan vaikutuksesta ohjelmistokehitykseen ja miten se vaikutti projektin tuotantoon, sisältäen huomioita työskentelymetodeista ja niiden käytöstä projektin kehityksessä. Kyseisen yrityksen vaikutus näkyi työnteossa vapaana ilmapiirinä. Kahden henkilön projektiryhmä osoittautui helposti hallittavaksi ja kommunikaatio oli sujuvaa. Ainoana ongelmana esiintyi projektin jäsenten läsnäolo. Kyseinen asia ei haitannut projektityöskentelyä enempää, kuin käytetyn projektinhallinnan viitekehyksen sääntöjen venyttämisenä. Lisäksi opinnäytetyössä läpikäytiin miten Suomen It-ratkaisut hyödyntää markkinarakoja suorittamaan omaa liiketoimintaansa ja miten se toimii käytännössä pitkäikäisten palveluiden ylläpidossa. Kyseinen projekti osoittautui hyväksi esimerkiksi valmiiksi olevan palvelun kehittämiseksi ja mitä siinä pitää ottaa huomioon tämänkaltaisen yrityksen kanssa. Työsuhde projektin omistajan kanssa osoittautui helposti lähestyttäväksi verrattuna edellisiin työkokemuksiini muissa projekteissa. Asiakassuhteet tuntuivat etäisemmiltä kiitos auktoriteetin ja informaationkulun järjestyksen.

Opinnäytetyön aikana suoritettiin projektin suunnittelu, tuotanto ja testaus loppuun asti. Projektin suunnittelun aikana perehdyimme, että miten huolenaiheiden erottaminen vaikutti järjestelmän rakenteen järjestämiseen ja sen positiiviseen vaikutukseen palvelun kehityksen aikana. Tämä valinta ohjelmiston arkkitehtuurin suunnittelusta todisti olevan hyvin kehittäjäystävällinen, koska se mahdollisti projektin laajentamisen selkeästi ja hallitvasti. Vähentäen metodien ja järjestelmien sekoamisen riskiä.

Tietoturvasta läpikäytiin pikaisesti tiedot mitä projektin tietoturvan toteutuksessa tarvittiin. Erityisesti keskittyen owasp tietolähteen hyödyntämiseen XSS hyökkäyksien estoon ja virheiden käsittelyn toteuttamiseen. Owasp todisti itsensä olevan oivallinen tiedonlähde, kiitos sen avoimen ja voittoa tavoittelemattoman toiminnan ansiosta. Tietolähdettä oli helppo konsultoida sen monipuolisten yhteisön kirjoittamien oppaiden ja tutkimusten ansiosta. Jokaiseen tietoturvan uhkaan löytyi monta perusteltua suojautumiskeinoa, jotka puolustivat Kulku www-sovellusta tietoturva uhkia vastaan.

Projektin aikana havaittiin erikoista tietoa tilanteista, jossa piti luoda järjestelmä, joiden piti kommunikoida toisten järjestelmien kanssa. Huomasimme pääongelman, miksi tämänkaltaisen järjestelmä pitää olla olemassa ja miksi siinä saattaa olla jotain pitkälle

tavoittelevaa ongelmaa ohjelmistokehityksen kannalta. Tätä kommunikaation vaikeutta mietiskeltiin ja pohdittiin. Lopullisesti teimme johtopäätöksiä sen alkuperästä. Samanlaisesti analysoin mistä tämä niin sanottu ongelma johtui, miten sitä voisi lievittää ja mitä hyviä sekä huonoja puolia sen lievittämisestä voisi syntyä. Ongelman syy teorioitiin olevan standardin puute toiminnanohjausjärjestelmien tiedonhallinnan kanssa. Kyseinen ajattelulinja toteutui tuloksettomaksi, mutta lisätutkimuksilla saattaisi olla mahdollisuus keksiä tapa kehittää kyseistä ohjelmistotuotannon vaihetta tehokkaammaksi.

Viimeistelyn aikana sain toteuttaa raportointityökalun ulkoasun ja tein havaintoja, miten järjestelmän muutokset vaikuttivat kyseisen ulkoasun toteutukseen. Järjestelmän muutokset vaikuttivat ulkoasuun, koska muutokset vähensivät hajanaisuutta huolenaiheiden erottamisen ansiosta. Tarpeettoman toiston poistaminen ja avaintoimintojen järjestäminen loivat yhden sivun järjestelmän, missä käyttäjälle käytettävissä olevat raporttimuodot olivat dynaamisesti luotu suoraan tietokannasta haettuina. Lisäksi ulkoasun suunnittelussa ja toteutuksessa otin huomioon vanhojen käyttäjien kokemukset vanhan käyttöliittymän kanssa ja sovelsin niin sanottuja hyviä osia, kun päivitin ulkoasun kuvastamaan uusia järjestelmän toimintoja. Tehdyt suunnittelupäätökset minivoivat edellisten käyttäjien sovelluksen uudelleen oppimiseen käytettyä aikaa.

Viimeistelyn loppupuolella asiakkaiden ja projektinomistajan palautteen mukaan raportointityökalua pitäisi tulevaisuudessa kehittää enemmän. Huomionarvoinen lisäys Kulku www-sovellukseen olisi ollut mahdollisuus luoda alaversioita valmiiksi tehdyistä raporteista. Kyseiseen muutokseen ei harmillisesti ollut tarpeeksi aikaa projektin loppupuolella.

Mielestäni projekti onnistui ratkaisemaan esitetyn pääongelman sopivalla tavalla huomioon ottaen projektin monilukuiset muuttujat. Ajattelen myös, että projekti olisi saattanut onnistua paremmin, jos olisi ollut enemmän aikaa sisäistää projektin pääongelman alkuperä.

LÄHTEET

Copras Project. 2007. Standardization Guidelines for IST research projects interfacing with ICT standards organizations. Viitattu 25.11.2019 Saatavilla sähköisesti osoitteessa <https://www.w3.org/2004/copras/docu/D27.pdf>

Dijkstra, E. 1976. The Effective Arrangement of Logical Systems. Burroughs Corporation. Viitattu 13.11.2019 Saatavilla sähköisesti osoitteessa <http://www.cs.utexas.edu/users/EWD/transcriptions/EWD05xx/EWD562.html>

Manico, J; Saad, E; Maćkowski, J & Bailey, R. 2019. OWASP Cheat Sheet Series Project. Viitattu 22.11.2019 <https://cheatsheetseries.owasp.org> > Cross Site Scripting Prevention

Manico, J; Saad, E; Maćkowski, J & Bailey, R. 2019. OWASP Cheat Sheet Series Project. Viitattu 23.11.2019 <https://cheatsheetseries.owasp.org> > Error Handling

Nielsen, J. 1994. 10 Usability Heuristics for User Interface Design. Viitattu 27.11.2019 Saatavilla sähköisesti osoitteessa <https://www.nngroup.com/articles/ten-usability-heuristics/>

OWASP. Owasp™ Foundation. Viitattu 20.11.2019 www.owasp.org > Home

OWASP. Industry:Citations. Viitattu 21.11.2019 www.owasp.org > Citations

OWASP. Viitattu 22.11.2019 www.owasp.org > Positive security Model

Suomen It-ratkaisut 2019. Yritys. Viitattu 20.11.2019 www.suomenitratkaisut.fi > Yritys

Schwaber, K. & Sutherland, J. 2017. The Scrum Guide™, The Definitive Guide to Scrum: The Rules of the Game. Viitattu 10.11.2019 Saatavilla sähköisesti osoitteessa <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>

Shapiro, C. Varian, H. 2003. A Strategic Guide to the Network Economy. Viitattu 25.11.2019