



Käyttöliittymän kehitys Now Experience viitekehyksellä

Ari McConchie

2021 Laurea





Laurea-ammattikorkeakoulu

Käyttöliittymän kehitys Now Experience viitekehyksellä

Ari McConchie
Tietojenkäsittely
Opinnäytetyö
Huhti, 2021

Ari McConchie

Käyttöliittymän kehitys Now Experience viitekehyksellä

Vuosi

2021

Sivumäärä

32

Opinnäytetyö perustuu toimeksiantajan tarpeeseen tarjota asiakkailleen yhteinen portaali-käyttöliittymä, josta on pääsy toimeksiantajan asiakkailleen tarjoamiin IT-palveluihin. Tuosta tarpeesta johtuen toimeksiantaja aloitti projektin, jossa kartoitettiin eri vaihtoehtoja portaalin toteuttamiseksi. Pääasiallisiksi vaihtoehtoiksi valikoitui nykyisen asiakasportaalin käytön jatkaminen jatkokehityksen ja parannusten jälkeen, tai siirtyminen ServiceNow -järjestelmään.

Projektissa tehtiin sekä nykyiseen järjestelmään että ServiceNow:hun POC (Proof-Of-Concept) portaalitoteutukset, eli eräänlaiset demototeutukset, joilla voidaan osoittaa järjestelmien kyvykkyyksiä ja soveltuvuutta käyttötarkoitukseen. ServiceNow:n osalta POC portaalitoteutus tehtiin järjestelmän Service Portal viitekehyksellä, mutta samalla selvitettiin, voidaanko uudempaa Now Experience viitekehystä hyödyntää.

Opinnäytetyön tavoitteena on arvioida ServiceNow -järjestelmään siirtymisen kannattavuutta ottaen huomioon toimeksiantajan tarpeet, ja selvittää Now Experience viitekehysten soveltuvuutta portaalikehitykseen vertailemalla sitä tyypillisesti käytössä olevaan Service Portaliin.

Lähtötilanteen kartoitus ja tarpeiden määrittely tapahtui haastattelututkimuksen kautta, missä haastateltiin nykyisen asiakasportaalin loppukäyttäjiä. Haastattelun tuloksena saatiin kattava aineisto, josta kävi ilmi, että nykyisessä ratkaisussa on paljon ongelmia ja haastatte-luista saatiin myös suuri määrä kehitysehdotuksia portaalin käyttökokemuksen parantamiseksi.

ServiceNow -järjestelmään siirtymistä arvioitiin projektissa tehdyn SWOT-analyysin kautta, eli vertailemalla järjestelmien vahvuuksia, heikkouksia, uhkia ja mahdollisuuksia. Lopputulos vertailussa oli se, että ServiceNow lienee ylläpidon ja jatkokehityksen kannalta parempi vaihtoehto, mutta riskialttiimpi kustannusten näkökulmasta.

Opinnäytetyö sisältää myös dokumentaatiota Now Experiencellä tehdystä kehityksestä ja joh-topäätöksiä sen soveltuvuudesta sen nykytilassa. Kehityksen aikana kävi nopeasti ilmi, että viitekehys ei ole vielä riittävällä valmiusasteella, että se olisi nykyisin yleisimmin käytettyyn Service Portaliin verrattuna kannattava vaihtoehto. On suositeltavaa odottaa, kunnes ServiceNow mahdollistaa portaalikehityksen Workspace käyttöliittymän kehitykseen tarkoitetun UI Builderin kautta ennen kuin ryhdytään suunnittelemaan portaalikäyttöliittymien kehitystä viitekehksellä.

Ari McConchie

UI development with the Now Experience Framework

Year

2021

Pages

32

The thesis is based on the client's need to provide its customers with a common portal interface that provides access to the IT services provided by the client. Due to that need, the client started a project that examined different options for implementing the portal. The main options were to continue using the current customer portal after further development and improvements, or to switch to the ServiceNow system.

The project carried out POC (Proof-Of-Concept) portal implementations for both the current system and ServiceNow. The implementations were demos that could be used to demonstrate the capabilities and suitability of the systems for the intended use. For ServiceNow, the POC portal was created with the Service Portal framework, but at the same time it was investigated whether the newer Now Experience framework could be used.

The aim of the thesis project was to evaluate the viability of the transition to the ServiceNow system, considering the needs of the client, and to find out the suitability of the Now Experience framework for portal development by comparing it with the typically used Service Portal.

The initial situation was examined, and the requirements were determined through an interview survey with the end users of the current customer portal. As a result of the interviews, comprehensive data were obtained, which showed that there are many issues with the current solution. The interviews also provided many development suggestions for improving the user experience of the portal.

The transition to the ServiceNow system was assessed through a SWOT analysis carried out in the project, in other words, by comparing the strengths, weaknesses, threats and opportunities of the systems. The result of the comparison suggested that ServiceNow is probably a better option from a maintenance and development perspective, but riskier from a cost perspective.

The thesis report also includes documentation of the development made with Now Experience and conclusions about its applicability in its current state. During the development, it quickly became clear that the framework is currently not a viable option compared to the commonly used Service Portal. It is recommended to wait until ServiceNow enables proper support for portal development through the UI Builder, which is currently used for Workspace UI development, before planning to develop portal UIs with the framework.

Keywords: ServiceNow, Now Experience, Service Portal

Sisälllys

1	Johdanto.....	7
1.1	Työn tavoitteet ja rajaukset	8
2	Projektin kuvaus.....	9
2.1	Projektimalli	9
3	Tutkimusmenetelmät	10
4	Haastattelututkimus	10
4.1	Käyttäjäprofiilit	11
4.2	Käyttö.....	11
4.3	Ongelmat	12
4.4	Kehitysideat.....	13
5	Vertailu / Benchmarking	13
5.1	Järjestelmän ylläpito	14
5.2	Järjestelmän kehitys	14
5.3	Kustannukset.....	15
5.4	SWOT-vertailu	15
6	Service Portal toteutus	18
7	Now Experience käyttöliittymän toteutus	18
7.1	Miksi Web Components?	19
7.2	Kehitysympäristön alustaminen	19
7.3	Komponenttien kustomointi	20
7.4	Workspace näkymä.....	21
7.5	ServiceNow Applikaatio	22
7.6	Service Portalin ja Now Experiencen eroavaisuudet	23
8	Johtopäätökset	24
9	Keskeiset käsitteet	25
9.1	ITIL	25
9.2	ITSM	26
9.3	SaaS (Software as a Service).....	26
9.4	ServiceNow	26
9.5	OOB (out-of-the-box)	27
9.6	Portaali	27
9.7	Now Experience UI viitekehys	27
9.8	Now CLI	28
	Lähteet.....	29
	Taulukot	32

1 Johdanto

Tämä opinnäytetyö liittyy toimeksiantajan sisäiseen projektiin, jossa selvitetään eri järjestelmien ja viitekehysten soveltuvuutta uuden asiakasportaalin kehittämiseen. Tällä hetkellä toimeksiantajalla on käytössä asiakasportaali heidän itsensä toimesta rakennetussa järjestelmässään, mutta projektissa pyrittiin selvittämään, olisiko kannattavaa siirtyä nykyisestä toteutuksesta SaaS -tyyppiseen ServiceNow järjestelmään. SaaS on lyhenne sanoista Software as a Service (ks. Keskeiset käsitteet), ja tarkoittaa käytännössä sitä, että järjestelmä ei ole toimeksiantajan kehittämä tai ylläpitämä, vaan se hankitaan kolmannen osapuolen toimittajalta. SaaS -mallissa on omat hyötynsä ja haittansa, joita käsitellään myös tässä opinnäytetyössä.

Vaikka toimeksiantaja tarjoaa asiakkailleen tällä hetkellä palveluportaalia, niin osa asiakkaista haluaa silti käyttää omia palveluportaalejaan. Se hankaloittaa toimeksiantajan tavoitetta itsepalvelun ja automaatioiden hyödyntämisestä. Osa asiakkaista ei käytä lainkaan palveluportaalia, ja jotkut sen sijaan hyödyntävät esimerkiksi itsepalveluominaisuuksia vähäisesti. Asiakasportaalissa esiintyvät käytettävyyssongelmat ja sen hidas kehittäminen on aiheuttanut turhautumista sekä sisäisesti että asiakkaiden keskuudessa. Edellä mainituista syistä on todettu, että asiakasportaalin kanssa tarvitaan projekti, jossa arvioidaan vaihtoehtoisia teknologioita portaalin toteutukseen, sekä tarvittavat ominaisuudet ja toiminnallisuudet, jotka asiakasportaalista pitää löytyä. (Toimeksiantaja 2020.)

Alla on listattuna yleisiä tarpeita ja linjauksia, jotka tulee ottaa huomioon asiakasportaalissa:

- Portaalin täytyy olla yhteensopiva asiakkaiden palveluportaalien kanssa
- Automaation on toimittava
- Nykytilanteen lisäksi on otettava huomioon asiakkaiden mahdolliset tarpeet jatkossa, jotta voidaan erottautua joukosta markkinassa
- Uusien ominaisuuksien käyttöönotto portaalissa täytyy olla joustavaa, kustannustehokasta ja nopeaa
- Portaalilla täytyy pystyä käyttämään myös mobiililaitteilla
- Asiakaskeskeinen kehittäminen

(Toimeksiantaja 2020.)

Terminä asiakasportaali tarkoittaa tässä tapauksessa graafista käyttöliittymää, jonka loppukäyttäjät ovat tämän opinnäytetyön toimeksiantajan asiakasyritysten henkilöt. Asiakasportaalien kautta käyttäjät voivat esimerkiksi tehdä erinäisiä palvelupyyntöjä, lukea tietämyskannan artikkeleita tai ilmoittaa ongelmista. Nämä toiminnot liittyvät toimeksiantajan IT-palveluihin, joita toimeksiantaja tarjoaa asiakasyrityksilleen.

Projektissa tehtiin POC portaalitoteutukset sekä vanhalla järjestelmällä, että ServiceNow:n Service Portal ominaisuudella. POC, eli proof-of-concept tarkoittaa toteutusta, joka ei välttämättä tule käyttöön valmistuessaan, mutta jonka tarkoitus on demonstroida järjestelmän kyvykkyyksiä ja soveltuvuutta käyttötarkoitukseen.

Opinnäytetyössä viitataan toisinaan toimeksiantajan lisäksi myös ServiceNow tiimiin. Toimeksiantaja on se osapuoli, jonka käyttöön asiakasportaalia kehitettiin projektissa. ServiceNow tiimillä sen sijaan tarkoitetaan toimeksiantajayritykseen kuuluvaa tiimiä, joka erikoistuu ServiceNow kehitykseen ja konsultointiin erityisesti asiakkaiden ServiceNow ympäristöissä, ja jonka vastuulla on projektissa ServiceNow POC -toteutus.

Service Portal on ServiceNow tiimille tuttu työväline, jolla voidaan suhteellisen helposti luoda näyttäviäkin käyttöliittymiä, lisäksi ServiceNow mahdollistaa paketoitujen portaalitoteutusten vaivattoman hyödyntämisen. Paketoinnilla tarkoitetaan tässä sitä, että ServiceNow:ssa aiemmin tehty ratkaisu on otettu talteen, jotta se voidaan käytännössä kopioida johonkin toiseen ServiceNow instanssiin muutamalla napin painalluksella. Näissä tapauksissa ratkaisun tuki täytyy olla sellainen, joka ei sisällä asiakaskohtaisia konfigurointeja, vaan ne voidaan tehdä paketoitujen ratkaisun käyttöönoton jälkeen erikseen.

1.1 Työn tavoitteet ja rajaukset

Service Portal on jo melko kauan käytössä ollut viitekehys, mikä on huolenaihe siinä mielessä, että jos ServiceNow lähitulevaisuudessa ottaa käyttöön jonkin uuden teknologian portaalien kehitykseen, niin luultavasti Service Portalin tuki loppuu siihen. Näin kävi esimerkiksi silloin, kun Service Portal korvasi sitä ennen käytössä olleen portaalikehitykseen tarkoitettua viitekehystä ServiceNow:ssa. Uusin ServiceNow:n julkaisema käyttöliittymäkehitykseen tarkoitettu viitekehys on nimeltään Now Experience, ja vaikka se on tällä hetkellä tuettu vain hyvin rajatusti ServiceNow:ssa, niin on täysin mahdollista, että se tulee korvaamaan Service Portalin tulevaisuudessa. Tämän vuoksi opinnäytetyössä tutkitaan pääasiallisesti Now Experienceen soveltuvuutta käyttöliittymäkehitykseen Service Portalin ja olemassa olevan asiakasportaalijärjestelmän vertailun lisäksi.

Näiden edellä mainittujen järjestelmien vertailukohtia pyritään arvioimaan projektin osana tehdyn vertailututkimuksen tuloksia hyödyntäen. Lisäksi tutkimuksen tukena voidaan käyttää projektin suunnitteluvaiheesta tehdystä haastattelututkimuksesta kerättyä tutkimustietoa. Tutkimuksen toivottu lopputulos on tarjota toimeksiantajalle dokumentaatiota Now Experienceen käyttömahdollisuuksista, sekä esittää vertailukohtia sen ja muiden edellä mainittujen käyttöliittymäkehityksen vaihtoehtojen välillä. Onko mahdollista tuottaa asiakasportaalin käyttöliittymä modernimpaa Now Experiencea hyödyntäen uuteen järjestelmään siirtyessä,

vai onko Service Portal parempi vaihtoehto? Kannattaako uuteen järjestelmään ylipäätään siirtyä, vai olisiko parempi kehittää olemassa olevaa portaalista niin, että sen käyttökokemus paranisi riittävästi?

2 Projektin kuvaus

Toimeksiantajan kanssa määriteltiin niin sanottu MVP (Minimum Viable Product) eli toteutus, joka täyttää toimeksiantajan vähimmäisvaatimukset lopputuotteelle. Se sisältää tyypilliset portaalitoiminnallisuudet, kuten Service Catalog, Knowledge ja Incident (ks. keskeiset käsitteet). Service Catalogin tulee tukea muun muassa asiakkaan määrittelemiä ja olemassa olevassa asiakasportaalissa käytössä olevia palvelupyyntölomakkeita.

Suunnittelun tukena käytettiin nykyisen asiakasportaalien loppukäyttäjien haastatteluista kerättyjä tuloksia. Haastatteluista ilmeni esimerkiksi, että jotkin ominaisuudet nykyisessä portaalissa ymmärretään usein väärällä tavalla ja itse asiassa hankaloittavat käyttökokemusta sen parantamisen sijaan. Näiden havaintojen perusteella voitiin pyrkiä korjaamaan joitain virkoja tai puutteita uuden käyttöliittymän kehityksen myötä.

2.1 Projektimalli

Ohjausmalliksi projektissa valittiin perinteinen vesiputousmalli, joka koostuu kolmesta päävaiheesta, suunnittelu, toteutus ja testaus. Vaikka malli ei ole iteratiivinen, niin projektissa pyritään kuitenkin joustamaan tilanteissa, joissa havaitaan uusia kehityskohteita, joita ei tunnustettu projektin suunnitteluvaiheessa.

Pulkkasen (2021) artikkelin mukaan vesiputousmalli on oikea valinta silloin, kun kyseessä on mittava kertaluontoinen ohjelmistoprojekti, joka on selkeästi määritelty ennalta eikä siihen liity suuria epävarmuustekijöitä. Näin ollen vesiputousmenetelmä koettiin oikeaksi tähän projektiin, sillä vaatimukset voidaan määritellä riittävän tarkasti projektin alussa olemassa olevan asiakasportaalien nykytilaa ja havaittuja puutteita tarkastelemalla. Pulkkanen (2021) toteaa vesiputousmallin hyväksi puoleksi sen, että projektin aikataulu, budjetti ja resursointi voidaan arvioida ennalta tarkasti.

3 Tutkimusmenetelmät

Projektissa tehtiin päätös hyödyntää kvalitatiivisena tutkimusmenetelmänä haastattelututkimusta tiedon hankintaan portaalin loppukäyttäjien vaatimusten ja toiveiden kartoittamiseksi. Haastattelumuotona oli teemahaastattelut, jonka Näpärä määrittelee puolistrukturoiduksi menetelmäksi, jonka kysymysten tai aiheiden järjestys ei ole välttämättä ennakkoon päätetty, mutta käsiteltävät aihepiirit ovat valmiiksi selvillä. Puolistrukturoitu haastattelu tarkoittaa sitä, että valmiita vastausvaihtoehtoja ei ole, ja sitä kautta haastattelu on vapaampi kuin strukturoitu haastattelu. Teemahaastattelun yksi oleellinen ominaisuus on haastattelijan ja haastateltavan vuorovaikutus. (Näpärä 2017.)

Opinnäytetyössä sovellettiin benchmarking -tutkimusmenetelmää järjestelmien vertailua varten. Shopify:n tiivistelmässä kuvaillaan benchmarkingia prosessiksi, jossa mitataan oman toiminnan tehokkuutta johonkin muuhun organisaatioon, joka nähdään olevan luokassaan huipulla. Benchmarkingiin kuuluu tiedon keruu erinäisten mittarien avulla ja niiden vertaileminen oman ja vertailtavan tuotteen, palvelun tai organisaation välillä. (Shopify 2021.)

4 Haastattelututkimus

Toimeksiantaja suoritti projektin osana haastattelututkimuksen haastatteleamalla nykyisen asiakasportaalin loppukäyttäjiä. Haastattelututkimuksesta kerättyä aineistoa hyödyntämällä saadaan käsitys siitä, mitkä ovat nykyisen järjestelmän ja sen käyttöliittymän kehityskohteet ja miten ne voitaisi korjata. Teemahaastattelun aihepiireiksi valikoitui pääasiassa tiketit, sisältö, portaalin käyttö, tekniset ominaisuudet ja raportointi

Haastattelututkimukseen otettiin haastateltavaksi toimeksiantajan asiakasyritysten edustajia, jotka ovat myös nykyisen asiakasportaalin loppukäyttäjiä ja näin ollen heillä on kokemusta sen käytöstä. Haastateltavilta kysyttiin heidän mieliteitään ja näkemyksiään nykyisestä portaalista, kuten sen hyvät ja huonot puolet, sen toimivuus ja sisällön kattavuus. Loppukäyttäjiltä haluttiin kuulla myös toiveita tai kehitysehdotuksia, joita voitaisiin hyödyntää uutta ratkaisua kehittäessä. (Toimeksiantaja 2020.) Haastattelututkimus auttaa kartoittamaan nykytilannetta ja tarjoamaan kehitysideoita uusille toiminnallisuuksille.

Haastatteluun pyrittiin ottaa mukaan 2-3 haastateltavaa per asiakas ja haastattelun kestoksi arvioitiin 45-60 minuuttia. Haastattelu tapahtui käytännöllisistä syistä Skypen välityksellä ja siinä oli mukana haastattelijan lisäksi myös kirjuri, vaikka haastattelu myös nauhoitettiin. Toivottiin myös, että käyttäjä näyttäisi jonkin oikean käyttötapauksen nykyisessä portaalissa, jos mahdollista.

4.1 Käyttäjäprofiilit

Haastattelututkimuksen aikana selvitettiin, minkälaiset käyttäjäprofiilit haastattelussa on osallisena, eli kuvakset erilaisista tyypillisistä asiakasportaalin loppukäyttäjistä. Käyttäjäprofiilien määrittäminen auttaa ymmärtämään loppukäyttäjien demografioita ja minkälaisia tarpeita asiakasportaalin niillä voi olla. Alla oleva taulukko sisältää projektissa määritellyt käyttäjäprofiilit.

Kenttätyöläinen	<ul style="list-style-type: none"> • On liikkeellä kentällä, maastossa tai muualla kuin työaseman äärellä • Tarvitsee helpon pääsyn palveluun erinäisissä olosuhteissa sijainnista riippumatta
Toimistotyöläinen	<ul style="list-style-type: none"> • Chattailee esimerkiksi palaverien aikana • Pääasiallinen kommunikointi näppäimistön kautta • Käyttää enemmän hakutoimintoja kuin chattia saadakseen portaalista esim. ohjeita
Juttelija, selvittelijä	<ul style="list-style-type: none"> • Ei tykkää hakea itse ohjeita vaan kommunikoi mieluummin puhelimitse • Aiheet saattavat olla vaikeita ilmaista chatissa • Tuntee Service Deskin henkilöt ja näin ollen yhteydenoton kynnys alhaalla
Tehokäyttäjä	<ul style="list-style-type: none"> • Käyttää sekä chattia että puhelinta ottaakseen yhteyttä • Kiinnostunut raporteista ja tilastoista

Taulukko 1: Käyttäjäprofiilit

(Toimeksiantaja 2020.)

4.2 Käyttö

Asiakasportaalin käyttöön liittyvistä kysymyksistä kävi ilmi, että tiketteihin liittyvät toiminnot ovat käytetyimmät toiminnot nykyisessä asiakasportaalissa, ja niiden käyttö koettiin helpoksi.

Portaalista seurataan sekä omia että muiden tikettejä, ja automaattitikettejä syntyy joissain tapauksissa yhtä paljon kuin manuaalisia. (Toimeksiantaja 2020.)

Chat -ominaisuutta käytetään vaihtelevasti eri asiakkaiden välillä, jotkut käyttävät paljon ja toiset eivät lainkaan. Chat koetaan hyödylliseksi, kun asialla ei ole kova kiire tai halutaan ottaa yhteyttä muiden toimien yhteydessä, esimerkiksi palaverissa. Silloin kun asialla on kiire, niin soittaminen koetaan paremmaksi vaihtoehdoksi, koska siten saa nopeasti palvelua. (Toimeksiantaja 2020.)

Kehuja nykyinen portaali sai esimerkiksi siitä, että linkit eri palveluihin ovat helposti saatavilla, raportointi on toimivaa, ja portaalissa näkyvät ilmoitukset ovat hyviä (Toimeksiantaja 2020).

4.3 Ongelmat

Teknisestä näkökulmasta ongelmakohdiksi nykyisessä portaalissa nousi asiakashaastatteluiden pohjalta esimerkiksi se, että portaali vaatii toimiakseen VPN-yhteyden ja portaalissa on yleisesti havaittavissa epävakausta ja hitautta. Linkkejä on vaikeaa tai jopa mahdotonta jakaa muille, koska portaali ohjaa loppukäyttäjän portaalin etusivulle klikatessa linkkiä. Chatin käyttökokemuksesta huonontaa se, että siihen ei voi kirjoittaa mitään ennen kuin saadaan yhteys tukihenkilöön. Oikeuspyynnöissä ei näy hyväksyjää, eikä ole selvää, miten vaihdetaan hyväksyjä esimerkiksi lomatuurausten ajaksi. (Toimeksiantaja 2020.)

Tikettien käsittelyä vaikeuttaa se, että järjestelmä luo paljon automaattitikettejä ja ohjaa ne käyttäjille, jolloin jonkin tietyn tiketin löytäminen on haasteellista. Automaattinen tiketin luonti sähköpostista ei toimi halutulla tavalla silloin kun on pidempi mailiketju. Tiketin luominen toisen puolesta on haastavaa, jotkut toiminnallisuudet eivät tue muita kieliä kunnolla, ja jotkut käyttäjät eivät ole edes löytäneet tukipyyntö -toiminnallisuutta. (Toimeksiantaja 2020.)

Sisällössä todettiin myös parantamisen varaa. Ohjeet portaalissa eivät ole ajan tasalla, eikä niiden joukossa ole asiakaskohtaisia ohjeita. Ohjeiden teko on hidasta ja joillekin on myös epäselvää mistä niitä edes tehdään, ei koeta vaivan arvoiseksi. Joskus linkkien takaa ei löydy mitään sisältöä, tai se on epäselvää. Järjestelmän näyttämät pop-up ilmoitukset näyttävät käyttäjien mielestä joskus viruksen aiheuttamilta. (Toimeksiantaja 2020.)

Portaalien olemassaolon tietoisuus todettiin ongelmaksi, harvat tuntevat sen kaikkia kyvykkyyksiä ja monesti unohtuu, että se on edes olemassa, jos sitä ei käytä paljoa. Käyttäjäkannan ylläpidossa on parantamisen varaa, sillä kaikkia käyttäjiä ei löydy portaalista. Portaalien käytön oppiminen on koettu haastavaksi, lomakkeiden löytäminen ja käyttäminen tuntuu

vaikealta, ja lomakkeet tuntuvat sopivan paremmin teknisesti asiantunteville käyttäjille. Monet käyttäjät vaativat mobiilikäyttöliittymää, mutta portaali soveltuu vain työasemakäyttöön. Portaalilla on myös työpöytäsovellus, mutta sen ei koeta tuovan minkäänlaista lisäarvoa, eikä sen olemassaolosta usein tiedetä. (Toimeksiantaja 2020.)

4.4 Kehitysideat

Tikettien automatiikkaan kaivattiin esimerkiksi mahdollisuutta luoda tikettejä sähköpostin forward -toimintoa käyttäen, ja että asiakkaiden omissa järjestelmissä kirjatut tiketit kirjautuisivat automaattisesti myös toimeksiantajan portaalijärjestelmään. Kiireelliset tiketit pitäisi pystyä merkitsemään niin, että niitä kohdeltaisiin korkeammalla prioriteetilla. Tikettien luonnissa voitaisiin hyödyntää esimerkiksi QR-koodeja ja tiketteihin voisi tuoda automatiikkaa mikä helpottaa esimerkiksi oikeiden ohjeiden tai toimien liittämistä tiketeille. Tikettien seurantaan kaivattaisiin myös parannuksia, jotta halutut tiketit löytyisivät helposti. (Toimeksiantaja 2020.)

Ohjeiden ylläpitoprosessi tulisi haastateltavien mielestä olla säännöllistä ja järjestelmän pitäisi tukea prosessia, sen lisäksi että ohjeet integroituisivat eri järjestelmiin. Ohjeiden syöttäminen järjestelmään koettiin nykyisellään vaikeaksi, koska se tehdään Service Deskin kautta sen sijaan, että ohjeita voisi luoda itse. Ohjeiden hakemiseen toivottiin parannuksia ja esimerkiksi ehdotuksia yleisimmin haetuista ohjeista. (Toimeksiantaja 2020.)

Mobiilikäyttöliittymään toivottiin työpöytäkäyttöliittymän ominaisuuksia, kuten tikettien avaamista ja seuranta, sekä uutisia ja ilmoituksia. Selkeät ohjeistukset ja tietojen näkyvyys koettiin mobiilissa tärkeäksi, ja chat ominaisuuksia kaivattiin. (Toimeksiantaja 2020.)

5 Vertailu / Benchmarking

Tuominen (2016) kuvailee Benchmarkingia prosessiksi, jossa tunnistetaan, ymmärretään ja sovelletaan erinomaisia käytäntöjä, jotka ovat käytössä muissa organisaatioissa oman organisaation suorituskyvyn parantamiseksi. Tämä ei kuitenkaan kuvaa täysin tarkasti tämän opinnäytetyön yhteydessä tehtävää vertailutukimusta. Ensinnäkin tässä tapauksessa ei varsinaisesti tunnisteta erinomaisia käytäntöjä omaavaa järjestelmää, vaan järjestelmiä ja teknologioita verrataan lähtökohtaisesti tasavertaisina.

Ilmeinen eroavaisuus Tuomisen kuvaukseen on myös se, että tässä ei vertailla organisaatioiden toimintaa, vaan eri teknologioita käyttöliittymän kehityksessä. Itse asiassa sekä toimeksiantajan nykyisin käytössä oleva järjestelmä, että ServiceNow ovat molemmat jo käytössä toimeksiantajalla eri käyttökohteissa. ServiceNow:ta ei vaan ole hyödynnetty portaalialustana toimeksiantajan IT-palveluissa siinä määrin kuin nykyisin käytössä olevaa portaalaa.

Tässä tapauksessa oli haastavaa löytää vertailututkimukseen konkreettisia mittareita, joiden kautta saataisi suoria tilastoja vertailukohteiden välillä. Konkreettisena mittarina toimi järjestelmien kustannukset, mutta muut vertailukohtat, ylläpito ja kehitys ovat abstraktimpia ja perustuvat lähinnä pohdintaan tiedossa olevien seikkojen pohjalta.

5.1 Järjestelmän ylläpito

Järjestelmät ovat ylläpidon näkökulmasta lähtökohtaisesti jo erilaiset. Nykyinen järjestelmä on toimeksiantajan itsensä ylläpitämä, mikä tarkoittaa sitä, että esimerkiksi järjestelmän päivitykset ja palvelinten tietokantojen ylläpito on myös toimeksiantajan vastuulla. ServiceNow on sen sijaan lähtökohtaisesti pilvipohjainen SaaS-järjestelmä (Software as a Service), jossa kolmas osapuoli, tässä tapauksessa ServiceNow, huolehtii edellä mainituista ylläpidollisista asioista. ServiceNow saa säännöllisin väliajoin versiopäivityksiä, jotka sisältävät esimerkiksi korjauksia olemassaoleviin toiminnallisiin ja uusia ominaisuuksia, joita asiakkaat voivat halutessaan ottaa käyttöön.

Toimeksiantajan ServiceNow portaalit rakennettaisiin niin sanotun on-premise instanssin päälle, mikä tarkoittaa, ettei se olisi ServiceNow:n pilviympäristössä, vaan toimeksiantajan omilla palvelimilla. Tämän vuoksi toimeksiantaja ei pääsisi hyödyntämään aivan täysin SaaS-mallin etuja ja joiltain osin järjestelmän ylläpito muistuttaisi enemmän nykyistä mallia. ServiceNow:n julkaisemat versiopäivitykset saadaan kuitenkin siitä huolimatta asennettua, tuoden uusia ominaisuuksia ja korjauksia järjestelmään.

5.2 Järjestelmän kehitys

Vaikka ServiceNow on kolmannen osapuolen kehittämä ja ylläpitämä järjestelmä, niin ServiceNow on mahdollistanut järjestelmän melko vapaan kustomoinnin asiakkaiden tarpeita vastaavaksi. Service Portal mahdollistaa tämän erityisen hyvin. Se koostuu niin sanotuista widgeteistä, joihin kehittäjä voi kirjoittaa web-kehitykselle tyypillistä HTML, CSS ja Javascript syntaksia. Widgetit sijoitetaan Bootstrap viitekehukseen perustuvalla ruudukolla, ja Service Portal tukee myös AngularJS viitekehystä.

Service Portal mahdollistaa siis melko vapaan kehityksen asiakasportaalin osalta, mutta ServiceNow tuo myös mukanaan joitain rajoitteita. ServiceNow on rajoitettu Javascriptin ECMAScript 5 versioon lukuunottamatta uutta Now Experience viitekehystä. Vanhentunut ECMAScript versio tarkoittaa esimerkiksi sitä, että jotkin asiat koodissa täytyy tehdä monimutkaisemmin, koska niille tarkoitettuja funktioita ei ole käytettävissä vielä nykyisessä versiossa.

ServiceNow:n dokumentaatio sivusto kuvaa kattavasti tarjolla olevia ratkaisuja koodauksessa käytettävien rajapintojen dokumentaation lisäksi, mikä helpottaa kehitystä. Sen lisäksi ServiceNow:lla on Community sivusto, jossa kehittäjät voivat jakaa osaamistaan ja tietoa muille, helpottaen kehitystä edelleen.

5.3 Kustannukset

Henkilöstökustannusten lisäksi ServiceNow aiheuttaa kustannuksia lisenssimallinsa ja esimerkiksi mahdollisten integraatioiden tietoliikennekustannusten kautta. ServiceNow:n hinnoittelu perustuu lisenssikäyttäjien määrään, portaalin loppukäyttäjien ei tarvitse olla lisenssikäyttäjiä. ServiceNow:n lisenssimalli on valitettavan monimutkainen ja toisinaan epäselvä, mikä vaikeuttaa kustannusten arviointia ja kehitystä, sen sijaan vanhan järjestelmän kustannukset ovat jo tarkasti tiedossa.

Projektissa tehdyn kustannusarvion mukaan ServiceNow ratkaisussa olisi suuremmat kustannukset kuin nykyisessä järjestelmässä, vaikka arvioon ei laskettaisi uuden portaalin kehityskustannuksia. Tosin tarkkaa arviota lienee vaikea tehdä ennen kuin uusi järjestelmä saadaan pilottikäyttöön.

5.4 SWOT-vertailu

Projektissa tehtiin niin sanottu SWOT-analyysi kahdelle järjestelmälle, eli vahvuuksien, heikkouksien, uhkien ja mahdollisuuksien kartoittaminen. SWOT-analyysin tuloksia hyödyntämällä voidaan tehdä arviointia ServiceNow järjestelmään siirtymisen kannattavuudesta. Järjestelmiä vertaillaan esimerkiksi ylläpidon, kehityksen ja kustannusten näkökulmista. Alla oleviin taulukoihin on kuvattu kooste arvioinnin tuloksista.

ServiceNow

Vahvuudet	<ul style="list-style-type: none"> Toimeksiantajan käytössä on jo käytössä ServiceNow alusta ja sitä kautta kokemusta sen ylläpidosta ja kehityksestä
------------------	--

	<ul style="list-style-type: none"> • Toimeksiantajalla on sisäisesti käytettävissään myös menestyneen ServiceNow partnerin konsultteja • ServiceNow:n yhden kontaktipisteen lähestymistapa; mahdollista toteuttaa useiden eri palvelunhallinnan osa-alueiden toimintoja
Heikkoudet	<ul style="list-style-type: none"> • Osaaminen ServiceNow:n Maria DB & Maria SQL:stä • Toimeksiantajan tarve toteuttaa portaali on-premise tyypisestä
Mahdollisuudet	<ul style="list-style-type: none"> • Mobiili sovellus käytettävissä out-of-the-box, vaatii vaan konfiguroinnin • ServiceNow:n innovaatiot, kuten koneoppiminen, tekoäly, NLU (Natural-language understanding), jne. ja kehityksellä selvä roadmap • Ei tarvitse lisenssejä portaalin loppukäyttäjille • Hyvät sisäänrakennetut integrointikyvykkyydet ulkoisiin järjestelmiin
Uhat	<ul style="list-style-type: none"> • Resursointi saattaa olla haastavaa • Joudutaan sitoutumaan ServiceNow:n lisenssimalliin • Riski että vanhan järjestelmän heikkouksia tuodaan myös uuteen järjestelmään

Taulukko 2: ServiceNow järjestelmän SWOT-analyysi

(Toimeksiantaja 2020.)

Nykyinen asiakasportaali

Vahvuudet	<ul style="list-style-type: none"> • Helposti integroitavissa muihin järjestelmiin • Lokalisoitu järjestelmä, monta kieltä valmiiksi saatavilla • Olemassa oleva järjestelmä, ei tarvetta siirtymälle ja kustannukset valmiiksi tiedossa • Toimeksiantajan oma tuote, joten kehitykseen liittyvät päätökset ovat myös toimeksiantajan käsissä • Toimivaa automatisaatiota jo valmiina järjestelmässä
------------------	---

Heikkoudet	<ul style="list-style-type: none"> • Tuotteen omistaja puuttuu, tuotetta ei kehitetä aktiivisesti tällä hetkellä • Ei roadmappia • Vaatii integraatioiden tekemistä, integraatiot tarvitsevat erillisen väylän • Palvelukatalogi vaatii tuplaimplementoinnin
Mahdollisuudet	<ul style="list-style-type: none"> • Helppo integroida toimeksiantajan sisäisiin työkaluihin ja järjestelmiin • Mahdollisuus laajentaa tuotetta jatkossa, kehitys omissa käsissä • Toimiva autentikaattioratkaisu • Toimeksiantajan tekemiä ominaisuuksia voidaan upottaa portaaliin
Uhat	<ul style="list-style-type: none"> • Asiakkaat saattavat silti haluta käyttää omia palveluportaalejaan • Osaaminen ja jatkokehityksen visio voi olla puutteellista • Järjestelmän maine sekä sisäisesti että ulkoisesti asiakkaiden keskuudessa • Asiakkaiden data jää toimeksiantajan järjestelmään, kun asiakas vaihtaa toimittajaa, mahdollinen GDPR riski

Taulukko 3: Nykyisen järjestelmän SWOT-analyysi

(Toimeksiantaja 2020.)

Tuloksista päätellen ServiceNow:lla on etu nykyiseen järjestelmään verrattuna erityisesti kehityksen osalta tarjolla olevan osaamisen, valmiin mobiilisovelluksen ja järjestelmässä valmiina olevien muiden toiminnallisuuksien ansiosta. Nykyisessä järjestelmässä ei myöskään ole aktiivista kehitystä eikä selvää suuntaa, miten sitä aiotaan kehittää, vaikka kehitys onkin vapaampaa kuin ServiceNow:ssa.

Nykyinen järjestelmä sen sijaan on kustannusten näkökulmasta vähemmän riskialtis, sillä kustannukset ovat jo valmiiksi tiedossa, eikä toimeksiantajan tarvitse sopeutua kolmannen osapuolen hinnoittelujärjestelmään. Globaalien asiakkaiden kanssa toimiminen lienee myös helpompaa nykyisessä portaalissa hyvän lokalisaation vuoksi.

6 Service Portal toteutus

Now Experience on tuore teknologia, jolla kehitystiimi ei ole aiemmin tehnyt minkäänlaista ServiceNow kehitystyötä, eikä sen soveltuvuudesta ollut vielä tietoa, joten asiakasportaalin kehityksessä tehtiin jo alkuvaiheessa päätös toteuttaa POC portaali olemassa olevaa Service Portal teknologiaa. Toimeksiantaja odottaa saavansa projektin päättyessä jonkinlaisen esitetävän lopputuotteen portaalitoteutuksesta, eikä projektitiimi voinut olla varma projektin alussa siitä, että voidaanko uutta Now Experiencea hyödyntää, vai olisiko vanhan Service Portal viitekehysten käyttäminen ainoa vaihtoehto. Tämän vuoksi toimeksiantajalle päätettiin tehdä toteutus pääasiallisesti Service Portal teknologialla hyödyntäen jo projektitiimin paketoitua Service Portal toteutusta.

Service Portal toteutuksen paketoinnilla tarkoitetaan sitä, että toteutus on tehty jo aiemmin ja tallennettu ServiceNow update settien muodossa paketointityökaluun, josta sen voi nopeasti ja helposti asentaa asiakkaiden ServiceNow ympäristöihin. Tämän vuoksi paketointiratkaisu on hyvä varasuunnitelma, koska sen toteutus vaatii vain hieman asiakaskohtaisia kustomointeja. Now Experience toteutus täytyisi sen sijaan tehdä alusta lähtien ja vaatisi siten huomattavasti suuremman työmäärän.

Verrattuna niin sanottuun out-of-the-box Service Portaliin, paketoitussa ratkaisussa on minimalistisempi graafinen käyttöliittymä, joka skaalautuu hyvin myös käyttöön mobiililaitteilla. Teknisestä näkökulmasta koodi on myös huomattavasti yksinkertaistettu out-of-the-box koodiin verrattuna, minkä vuoksi mahdollisten kustomointien tekeminen on huomattavasti helpompaa.

Service Portal toteutusta ei saatu vielä valmiiksi, mutta se ehdittiin esitellä projektin ohjausryhmälle. Portaali sai kehuja muun muassa ServiceNow:n tietämyskannan ja palvelupyyntöjen käyttöliittymien toiminnoista, kuten yleisimpien tai parhaiten arvioitujen ohjeiden näytöistä.

7 Now Experience käyttöliittymän toteutus

Siinä missä Service Portal ja nykyisen portaalin kyvykkyydet olivat tiedossa jo projektin alussa, Now Experience kehitys vaati teknologian opiskelua, ennen kuin sillä voitaisi kehittää minkäänlaista ratkaisua. Oli myös epäselvää, voidaanko sitä ylipäätään hyödyntää tämän

projektin tavoitteiden saavuttamiseksi. Tämän vuoksi tämä osio keskittyy viitekehysten tekniikkiin valmiuksiin ja soveltuvuuteen.

Now experience viitekehys Service Portalista poiketen Web Componentsiin. Web Components on kokoelma teknologioita, joilla voidaan luoda web kehityksessä käytettäviä kustomoituja elementtejä, joiden toiminnallisuudet ovat kapseloituja muusta koodista konfliktien välttämiseksi (Mozilla 2021).

7.1 Miksi Web Components?

Web Components ei vaadi toimiakseen viitekehystä, kuten Reactia, mikä tekee siitä kevyen ja hyvän suorituskyvyn kannalta (Kamboj 2020). Service Portalissa on sen sijaan käytössä AngularJS viitekehys, joten on mahdollista, että Now Experienceen suorituskyky olisi parempi Web Componentsin ansiosta.

Wedekind kirjoittaa artikkelissaan, että koodia ei kirjoiteta vain koneiden ymmärrettäväksi, myös muiden kehittäjien. Perinteiseen HTML:ään verrattuna Web Components mahdollistaa helpommin luettavan koodin kirjoittamisen ja tehokkaamman kehityksen esimerkiksi sen uudelleen käytettävien komponenttien ja Shadow DOMin ansiosta. (Wedekind 2018.)

Organisaation näkökulmasta Web Components parantaa kustannustehokkuutta, koska kehittäjät voivat säästää aikaa käyttäen uudelleenkäytettäviä toiminnallisuuksia sen sijaan, että kaikki täytyisi kehittää puhtaalta pöydältä. Toiston vähäisyys ohjelmoinnissa myös helpottaa web applikaation brändäystä. (Wedekind 2018.)

Yksi Web Componentsin merkittävistä eduista on myös selaintuki. Custom Elements v1:n ansiosta kehittäjät voivat luottaa, että useimmat ominaisuudet toimivat nykyaikaisilla selaimilla. Web Componentsille on myös useita määriteltyjä parhaita käytäntöjä, joita noudattamalla kehitystä voidaan yhdenmukaistaa tiimien välillä. (Wedekind 2018.)

7.2 Kehitysympäristön alustaminen

Tyypillisestä ServiceNow kehityksestä poiketen Now Experience viitekehys vaatii joitain komponentteja asennettaviksi paikalliseen muistiin. Kehitys itsessään tapahtuu Now CLI komentorivin komentojen avulla, ja komentorivin asennus vaatii puolestaan node.js:n ja npm:n asennuksen toimiakseen.

Ennen kuin kehitys voidaan aloittaa, täytyy ensin asentaa npm, joka on yksi Now CLI:n riippuvuuksista. Ensimmäiseksi täytyy tehdä globaali npm asennus

```
>npm install --global @servicenow/cli
```

komennolla. Sen jälkeen täytyy nytään now experience projektin kansioon cd komennolla ja tehdään npm asennus

```
>npm i
```

komennolla.

Näin saadaan tarvittavat riippuvuudet käytettäväksi projektille ja kehitys voidaan aloittaa.

Npm:n asennuksen jälkeen voidaan luoda web component projekti Now CLI:n project komennolla:

```
>now-cli project --name @mytest/hello-world --scope x_mytest_helloworld --offline
```

Project komennon parametrina annetaan komponenttia kuvaavan nimen lisäksi scope parametri, joka määrittää mihin applikaatioon projekti sijoitetaan silloin, kun se vietään ServiceNow kohdeinstanssiin. Project komennon tuloksena paikalliseen muistiin syntyy tarvittavat tiedostot, jotka voidaan avata IDE:ssä ja kehitys voidaan aloittaa.

7.3 Komponenttien kustomointi

Vertaillen Now Experiencea Service Portaliin oli oleellista selvittää, voidaanko jo olemassa olevia Workspacen komponentteja kustomoida. Näin voitaisi säästää huomattavasti aikaa verrattuna siihen, että kaikki komponentit luotaisiin alusta saakka itse. Jos se ei ole mahdollista, olisi ainakin tärkeää saada riittävä dokumentaatio komponenttien kehityksestä.

Custom elementtien kehitystä voitaisi nopeuttaa huomattavasti, jos järjestelmän vakiokomponenttien lähdekoodi voitaisi kopioida kehittäjän IDE:een ja tehdä siellä tarvittavat muokkaukset kustomoituja toiminnallisuuksia varten. Näin tehdään useimmiten Service Portal widgettien kehityksessä, koska koodin kirjoittaminen alusta asti itse kestäisi hyvin kauan verrattuna olemassaolevan koodin muokkaukseen. Tämän vuoksi projektissa yritettiin löytää ServiceNow:n tietokannasta taulu, johon komponentit tallennetaan, jotta komponenttien koodia voidaan tarkastella. Taulu löydettiin, mutta valitettavasti itse komponenttien koodi on piilotettu käyttäjiltä, joten ainakaan ServiceNow käyttöliittymästä ei voida tehdä komponenttien koodin kopioimista.

Elementtien lähdekoodi saatiin kopioitua lopulta käyttäen kolmannen osapuolen ohjelmaa. Ensinnäkin järjestelmästä täytyy löytää haluttu komponentti taulusta sys_ux_lib_component ja avata siihen liitetty source script. Source script tiedosto sisältää niin sanottua minimoitua Javascript syntaksia. Minimoidulla syntaksilla tarkoitetaan tässä tapauksessa sellaista koodia, josta on poistettu kaikki ylimääräinen tai turha data, ilman että se vaikuttaa ohjelman suoritukseen (Google 2018). Minimointi vähentää tiedonsiirtoa verkkosivujen latauksessa ja tehostaa verkkosivujen nopeutta, mikä ansiosta käyttäjäkokemus parantuu koodin minimoinnin

seurauksena (Imperva 2021). Koodin minimoinnilla on siis selvästi positiivinen vaikutus suorituskyvyn kannalta, mutta sitä on hyvin vaikea lukea ja ymmärtää kehittäjän näkökulmasta.

Koodaajat eivät tyypillisesti kirjoita syntaksia lyhimmissä mahdollisessa muodossa, vaan pyrkivät siihen, että koodi on selkeää ja helposti ymmärrettävissä muillekin kehittäjille. UX komponenttien source scriptin minimoitu Javascript syntaksi on suositeltavaa muuntaa luettavaan muotoon, ennen kuin kehittäjä ryhtyy muokkaamaan komponenttia.

Minimoidun koodin muuttaminen luettavaan muotoon onnistuu seuraavia vaiheita noudattaen:

1. Etsi oikea UX Component definition tietue, siirry sen Source Script tietueeseen ja klikkaa source script linkkiä, joka sisältää minimoidun javascript koodin.
2. Lisää ".map" source script sivun URL osoitteen loppuun, selaimen pitäisi tässä vaiheessa ladata minimoitu koodi paikalliseen muistiin.
3. Siirry SrcBrowse.com sivulle, jossa sijaitsee kolmannen osapuolen työkalu koodin muuntamiseen. Sivulle voi viedä ladatun source scriptin, jolloin ruudulla näkyy koodin ymmärrettävään muotoon muunnettu versio, jonka voi joko ladata tai tarkastella selaimessa.

(Semaphore Partners 2020.)

Vertailun kannalta on huomioitavan arvoista, että edellä mainittu prosessi järjestelmän vakio-komponenttien kustomoimisen aloittamiseksi on hyvin aikaa vievä verrattuna Service Portal widgettien kustomointiin, mikä vaatii vain napin painalluksen widget editorissa.

7.4 Workspace näkymä

Tällä hetkellä ServiceNow:n oman dokumentaation mukaan Now Experience on tuettuna ainoastaan Workspace näkymien kehityksessä (ServiceNow 2021). Sen vuoksi asiakasportaalien käyttöliittymää luodessa tarkasteltiin mahdollisuuksia muokata Workspace näkymää niin, että ulkoasu muistuttaisi tyypillistä portaalinäkymää. Toki todellisuudessa Workspacea ei luultavasti voitaisi hyödyntää näin, esimerkiksi lisenssirajoitusten vuoksi, mutta Now Experience portaalien voitaisi näin demonstroida toimeksiantajalle helpommin.

Tyhjässä Workspace näkymässä on sivun yläosassa header-palkki, joka sisältää esimerkiksi yrityksen logon, käyttäjän profiilivalikon ja hakutoiminnon. Sen lisäksi ruudun vasemmassa laidassa on valikko, josta voidaan selailla Workspacen eri sivuja. Parhaassa tapauksessa nämä osiot saataisi joko piilotettua täysin tai kustomoitua niin, että saataisi aikaiseksi portaalinäkymä, mutta saataisiin kuitenkin hyödynnettyä Workspacen kustomointityökaluja, kuten UI builderia.

Workspacelle voidaan määritellä yrityksen brändin mukainen väriteema ja logo, mutta valitettavasti asetukset pätevät vain header-osioon. Tämän vuoksi sivuvalikko pyrittiin piilottamaan näkymästä, koska sen väriteema ei sopinut määriteltyyn teemaan. Sivuvalikon piilottaminen kuitenkin piilotti myös sivujen sisällöt täysin, joten se ei käytännössä ollut mahdollista.

Workspacen UI builder todettiin riittävän tehokkaaksi työkaluksi sivun asetteluun, mutta näkymän ulkoasurajoitteiden vuoksi portaalinäkymän kehittäminen Workspacessa ei liene realistista. Lisäksi ServiceNow:n lisenssimalli tekee tästä vaihtoehdosta kyseenalaisen, koska Agent Workspace on tarkoitettu lisenssikäyttäjille, jotka esimerkiksi käsittelevät loppukäyttäjien tekemiä tikettejä.

7.5 ServiceNow Applikaatio

Workspace näkymien rajoitteiden vuoksi selvitettiin, voidaanko Now Experiencella luotuja komponentteja hyödyntää muualla järjestelmässä niin, että rajoitteista päästään eroon. Tähän tarkoitukseen luotiin kustomoitu ServiceNow applikaatio, joka sisältäisi portaalinäkymän. Tällä tavalla saadaan tehtyä kehitystä vapaammin, koska käyttöliittymä rakennetaan niin sanotusti puhtaalta pöydältä.

ServiceNowssa luotu web applikaatio koostuu viidestä taulusta, jotka muodostavat applikaation rakenteen:

- **UX Registry** (`sys_ux_page_registry`): Säiliö web applikaatiolle. Sisältää applikaation URL osoitteen.
- **UX Page** (`sys_ux_page`): Ylätason UI container, jonka tarkoituksena on renderöidä root element komponentti, joka määrittää web applikaation käyttöliittymän asettelun.
- **UX Page Element** (`sys_ux_page_element`): Container elementti komponentille tai muille UX Page Elementeille.
- **UX Component Definition** (`sys_ux_lib_component`): Wrapper web componentin varsinaiselle koodille.
- **UX Source Code** (`sys_ux_lib_source_script`): Web componentin Javascript koodi. (Pishchulin 2020.)

Näistä UX Registrylle, UX Pagelle ja UX Page Elementille täytyy luoda ACL säännöt, jotta niihin voidaan luoda tietueita. Kun ACL säännöt ja taulurakenteen tietueet on luotu, niin kehittäjä voi ryhtyä luomaan omia kustomoituja komponentteja käyttäen Now CLI:tä ja viedä ne tähän ServiceNow:ssa tehtyyn web applikaatioon. (Pishchulin 2020.)

Tämän ratkaisun testaamista varten luotiin yksinkertaisia kustomoituja komponentteja ja sijoitettiin yllä mainittuun taulurakenteeseen. Ilman Workspacessa käytössä olevaa UI Builderia tai Service Portalin Page Designeria lienee kuitenkin haastavaa tehdä sivun asettelusta haluttu näköinen, kun sivulle asetettiin useampi komponentti. Sen takia tämä ratkaisu todettiin vaivalloiseksi verrattuna muihin tarjolla oleviin vaihtoehtoihin, lisäksi kustomoidun web applikaation kanssa voi olla haastavaa rajoittaa käyttäjien oikeuksia erityisesti Now Experiencen komponenttien datanhallinnan puutteen vuoksi.

7.6 Service Portalin ja Now Experiencen eroavaisuudet

Yksi mainitsemisen arvoinen asia näiden kahden viitekehyksen eroista on se, että kehittämissä käytettävissä olevat työkalut ovat huomattavasti helppokäyttöisemmät Service Portalilla. Ensinnäkin Service Portal ei vaadi minkäänlaista asennusta paikalliseen muistiin, vaan kaikki tarvittava löytyy pilviympäristöstä. Tämä sallii kehittämistyön tekemisen millä tahansa työasemalla, jossa on internet-yhteys ja tuettu verkkoselain. Now Experience sen sijaan vaatii Now CLI:n asentamisen tietokoneelle.

Vaikka Now CLI on sinänsä yksinkertainen komentoliittymä, niin sen käyttöä vaikeuttaa esimerkiksi se, että virheviestien sisältö on usein epämääräinen, ja niistä on vaikea selvittää ongelman juurisyitä. Yksi esimerkki epämääräisestä virheviestistä, jonka sain ajaessani login komennon on **Instance version "Orlando" unsupported!**. Virheviesti on erikoinen siinä mielessä, että Orlando on ServiceNow:n oman dokumentaation mukaan Now Experiencessä tuettu ServiceNow versio, ja viitekehyksen julkaisun aikaan ainoa tuettu versio (ServiceNow 2020).

Now Experiencen eduksi voidaan huomioida se, että se pohjautuu uudempaan Javascriptin ECMAScript versioon kuin ECMAScript 5, joka on tällä hetkellä ServiceNow:n uusin tuettu versio muissa skriptoissa, mukaan lukien Service Portalissa kirjoitettavat skriptat. Tämä ilmenee esimerkiksi siitä, että Now Experiencen komponentteja kirjoittaessa voidaan alustaa const ja let tyyppisiä Javascript muuttujia. Const ja let muuttujia EcmaScript 5 ei tue, koska muuttujatyypit lisättiin vasta ECMAScript 6:ssa (Orendorff 2015).

Uudempi Javascript versio auttaa esimerkiksi siinä suhteessa, että voi olla helpompaa löytää ohjeita tai apua Javascript ohjelmointiin netistä, koska mille tahansa ECMAScript versiolle suunnattuja ohjeita voidaan hyödyntää. Sen lisäksi ohjelmoija, joka on aikaisemmin ohjelmoinut esimerkiksi ECMAScript 6 Javascriptiä, pystyy helpommin siirtymään ServiceNow kehitykseen, koska hänen ei silloin tarvitse opetella käyttämään vanhempaa Javascript syntaksia.

Workspacessa sivujen asettelua varten on käytössä työkalu nimeltään UI builder. Tuota työkalua verrattiin projektissa esimerkiksi Page Designeriin, joka on sen sijaan Service Portalin sivuasetteluun käytettävä työkalu.

8 Johtopäätökset

Paikallisesta on-premise ratkaisusta huolimatta ServiceNow tarjoaa ylläpidollisesta näkökulmasta joitain etuja verrattuna nykyisin käytössä olevaan järjestelmään. Säännöllisesti julkaistut versiopäivitykset tuovat mukanaan uusia ominaisuuksia ja korjauksia nykyisiin toiminnallisuuksiin ja päivitysten asennusprosessi on suurimmaksi osaksi automaattinen.

Kehityksen näkökulmasta siirtyminen ServiceNow -alustaan on kannattava siksi, että toimeksiantajalla on sisäisesti hyvin paljon ServiceNow osaamista käytettävissään. ServiceNow tarjoaa monia valmiita toiminnallisuuksia, jotka vaativat vain konfigurointia tai pieniä kustomointeja, mikä vähentää kehityksen työtaakkaa. Lisäksi ServiceNow tarjoaa kattavan dokumentaation ja Community -sivuston, josta löytyy apua ServiceNow kehitykseen ja konfigurointiin.

Kustannusten näkökulmasta siirtyminen ServiceNow järjestelmään on riskialttiimpi vaihtoehto kuin pysyä nykyisessä portaaliratkaisussa. Sen lisäksi että monimutkainen lisenssimalli hankaloiittaa kustannusten arviointia, niin malliin saattaa tulla tulevaisuudessa myös muutoksia, joita ei voida varautua ennalta. Nykyisen järjestelmän ylläpidon kustannukset ovat sen sijaan jo valmiiksi tiedossa.

Jo projektin alkuvaiheessa oli varsin selvää, että uudella Now Experience viitekehyksellä olisi tällä hetkellä hyvin haastavaa ja aikaa vievää tehdä loppukäyttäjille suunnattu portaalitoteutus. Tähän vaikutti ensinnäkin se, että uusi viitekehys on ainakin ilman hyvin vahvaa kustomointia käytettävissä vain ja ainoastaan Agent Workspace käyttöliittymän sisällä. Agent Workspace ei nimensä mukaisesti ole tarkoitettu loppukäyttäjien tarpeisiin, vaan se sisältää toiminnallisuuksia, joita esimerkiksi palvelupyntöjen käsittelijät tai service desk agentit käyttävät työssään. Projektissa tutkittiin mahdollisuutta kääntää Workspace näkymä portaalin kaltaiseksi, mutta huomattiin hyvin nopeasti, että Workspacen vakioasetteluja ja valikoita ei voida piilottaa tai muokata juuri lainkaan.

Agent Workspacen hyödyntämisen mahdollisuuden lisäksi projektissa selvitettiin, voidaanko ServiceNow ympäristöön tehdä applikaatio omalla käyttöliittymällään, jossa voidaan hyödyntää Now Experiencen custom componentteja. Se oli toki mahdollista, mutta tämän menetelmän soveltuvuus portaalien kehittämisessä on luultavasti erittäin heikko. Tässä lähestymistavassa aloitetaan täysin puhtaalta pöydältä siinä mielessä, että esimerkiksi Service Portalissa

ja Agent Workspacessa käytettävissä olevat asettelutoiminnallisuudet ja niihin liittyvät käyttöliittymät (UI Builder, Page Designer) puuttuvat täysin, ja ne pitäisi rakentaa alusta asti. Tämä veisi hyvin paljon aikaa verrattuna jo olemassa olevien ServiceNow toiminnallisuuksien hyödyntämiseen.

Now Experiencea ei siis voida tällä hetkellä hyödyntää portaalikehityksessä. Sillä kuitenkin keuhkeiltiin luoda joitain komponentteja jonkin yleisen portaalitoiminnallisuuden saavuttamiseksi. Uusi viitekehys saattaa tulla tulevaisuudessa käyttöön uusissa portaaaleissa, joten sen vuoksi komponenttien soveltuvuutta siihen yritettiin jo tässä vaiheessa selvittää. Lisäksi tästä voisi olla hyötyä jatkossa Agent Workspace toteutuksissa.

Yksi uuden viitekehysten hyödyistä on se, että se perustuu uudempaan Javascript ECMAScript versioon kuin Service Portal, tai ServiceNow yleensäkin. ServiceNow on jo pitkän aikaa tukenut vain ECMAScript 5:ttä, joka julkaistiin vuonna 2009. Järjestelmän tukema ECMAScript on siis tässä vaiheessa hyvin vanha, eikä tue monia ominaisuuksia, jotka uudemmat versiot ovat tuoneet Javascriptissä hyödynnettäviksi.

Lopputuloksena voidaan todeta, että nykyisen asiakasportaalin korvaaminen ServiceNow -alustalle rakennetulla portaalilla on varteenotettava vaihtoehto, mutta Now Experienceen hyödyntäminen siihen ei ole tällä hetkellä toteuttamiskelpoinen ratkaisu. Service Portal sen sijaan tarjoaa riittävät kyvykkyydet asiakastarpeisiin vastaamiseen. Tämän opinnäytetyön kirjoittamisen aikana ServiceNow julkisti, että uudessa Quebec -versiossa uutta viitekehystä ja siihen liittyvää UI Builderia voidaan hyödyntää myös portaaliratkaisujen kehittämisessä (ServiceNow 2021). Jatkossa siis lienee kannattavaa perehtyä syvemmin Now Experienceella kehittämiseen.

9 Keskeiset käsitteet

Opinnäytetyössä esiintyvät termit ja käsitteet on pyritty selventämään niiden esiintyessä tekstissä, mutta alla on listattuna vielä tarkemmat määritelmät siltä varalta, että lukija tarvitsee lisätietoa jostakin opinnäytetyön aihepiiristä.

9.1 ITIL

IT Infrastructure Library, eli ITIL on viitekehys, joka sisältää parhaita käytäntöjä IT palveluiden toimittamiseen. ITIL:iä on kehitetty vuosien varrella ja tällä hetkellä koostuu viidestä kirjasta, joista jokainen kuvaa erinäisiä IT palveluiden elinkaareen kuuluvia prosesseja ja vaiheita. (White & Greiner 2019.)

ITIL:n lähestymistapa IT-palvelunhallintaan voi auttaa yrityksiä riskienhallinnassa, asiakassuhteiden vahvistamisessa ja kustannustehokkuuden parantamisessa (White & Greiner 2019).

Axeloksen mukaan ITIL on maailmanlaajuisesti yleisimmin omaksuttu ohjeistus IT-palvelunhallinnassa. Sitä voidaan hyödyntää riippumatta siitä, onko kyseessä pieni tai suuri, paikallinen tai kansainvälinen, tai julkisen tai yksityisen sektorin organisaatio. (AXELOS 2013.)

9.2 ITSM

ITSM, eli IT-palvelunhallinta viittaa kaikkiin aktiviteetteihin, mitkä liittyvät siihen, miten organisaatiot tarjoavat IT-palveluita asiakkaille tai loppukäyttäjille. Tähän sisältyy palvelun suunnittelu, toimitus, käyttö sekä hallinta. (ITILTRAINING 2016.)

9.3 SaaS (Software as a Service)

SaaS:lla tarkoitetaan ohjelmiston toimitusmallia, jossa dataan pääsee käsiksi millä vain laitteella, jossa on verkkoselain ja internet-yhteys. Tässä toimitusmallissa ohjelmiston toimittaja hostaa ja ylläpitää palvelimia, tietokantoja ja koodia, joista ohjelmisto koostuu. (Short 2020.)

SaaS poikkeaa perinteisestä on-premise mallista kahdella merkittävällä tavalla:

- SaaS asennukset eivät vaadi asiakkaalta mittavaa laitteistoa, ja vastuu sen ylläpidosta ja huollosta voidaan ulkoistaa.
- SaaS järjestelmät maksetaan tyypillisesti jatkuvan tilauksen muodossa, sen sijaan, että kustannukset tapahtuisivat kertamaksuna.

(Short 2020.)

9.4 ServiceNow

ServiceNow on pilvipohjainen SaaS (Software as a Service) järjestelmä, johon sisältyy muun muassa ITSM (IT services management), ITOM (IT operations management) and ITBM (IT business management) kyvykkyksiä. (Fitzgibbons 2020). Yleisimmät järjestelmän käyttötapaukset kuitenkin keskittyvät järjestelmän ITSM toiminnallisuuksiin. ServiceNow on suunniteltu ITIL-viitekehyksen pohjalta (ServiceNow 2020).

Tyypillisiä käyttötapauksia järjestelmälle on esimerkiksi organisaation help deskin tai service deskin toiminnan organisointi, laajamittaisten projektien hallinta ja benchmarkkaus erinäisten työkulkujen edistymisen mittaamiseen ja mallinnukseen (Fitzgibbons 2020).

ServiceNow sisältää myös useita työkaluja, joilla voidaan integroida ServiceNow instanssi kolmannen osapuolen järjestelmiin. Esimerkiksi ServiceNow:n Cloud Management sovelluksella voidaan integraation kautta hallinnoida Amazon Web Services, Microsoft Azure tai VMware resursseja yhdestä käyttöliittymästä (ServiceNow 2021).

9.5 OOB (out-of-the-box)

ServiceNow on kolmannen osapuolen tuote, joka sisältää valmiiksi useita toiminnallisuuksia, mutta tyypillisesti siihen tehdään asiakaskohtaisesti erinäisiä kustomointeja. Out-of-the-box tämän opinnäytetyön yhteydessä viittaa ServiceNow -alustaan tai sen toiminnallisiin siinä tilassa, missä ne ovat ennen kuin niihin on tehty kustomointeja. Tähän määrittelyyn ei lasketa esimerkiksi asiakaskohtaista dataa, kuten organisaatio- tai käyttäjätietoja, joita on tuotu järjestelmän tietokantaan.

9.6 Portaali

Tämän opinnäytetyön kontekstissa termi portaali tarkoittaa ServiceNow -järjestelmän graafista käyttöliittymää, jonka kautta loppukäyttäjät voivat esimerkiksi tehdä palvelupyyntöjä, saada tukea tai ilmoittaa ongelmista.

Portaalit rakennetaan tällä hetkellä tyypillisesti Service Portal viitekehysellä, joka on modernimmille teknologioille pohjautuva vaihtoehto vanhalle Content Management System (CMS):lle (ServiceNow 2021).

9.7 Now Experience UI viitekehys

Now Experience UI viitekehys on kehitetty tarjoamaan moderni ja puhdas tapa ServiceNow kehittäjille luoda käyttökokemuksia Now alustalla. Se pohjautuu uusimpiin web standardeihin, kuten Web Components ja sen on tarkoitus olla pitkäkestoinen kehitysväline. (ServiceNow 2020.)

9.8 Now CLI

Now CLI (Command Line Interface) on Now Experience kehityksessä vaadittava komentorivi, jota käytetään projektin luomiseen, komponenttien testaamiseen localhostissa, yhteyden muodostamiseen kohdejärjestelmän kanssa, ja valmiiden komponenttien lähettämiseen kohdejärjestelmän tietokantaan.

Kyseessä on melko yksinkertainen komentorivi, kuten sen käytettävissä olevien komentojen listasta näkyy:

```
Commands:
  login [host] [method] [username]
    [password]                               Login to a glide instance.
    deploy                                     [aliases: connect, conn]
                                             Deploy a component to a glide
                                             instance.
                                             [aliases: gd]
  logout [host]                               Logout from a glide instance.
                                             [aliases: disconnect, dis]
  project [options]                           Creates and scaffolds a new project
                                             with all necessary files to start
                                             development.
                                             [aliases: project]
  develop [options]                           Run component in local development
                                             environment.
                                             [aliases: dev]

Options:
  --help      Show help                       [boolean]
  --version   Show version number             [boolean]
```

Lähteet

Sähköiset

AXELOS. 2013. Essential ITIL: Processes and functions. Viitattu 20.3.2021. <https://ebookcentral.proquest.com/lib/laurea/reader.action?docID=4568791>

Fitzgibbons L. 2020. ServiceNow. Viitattu 20.3.2021. <https://searchitoperations.target.com/definition/ServiceNow>

Google, 2018. Minify Resources (HTML, CSS, and JavaScript). Viitattu 29.1.2021. <https://developers.google.com/speed/docs/insights/MinifyResources>.

Imperva. Minification. Viitattu 29.1.2021. <https://www.imperva.com/learn/performance/minification/>.

ITILTRAINING. 2016. The Difference Between ITIL and ITSM. Viitattu 20.3.2021. <https://www.itiitraining.com/eur/blog/difference-til-itsm>

Kamboj S. 2020. Web Components Basics and Performance Benefits. Viitattu 29.1.2021. <https://medium.com/@spkamboj/web-components-basics-and-performance-benefits-f7537c908075>

Mozilla, 2021. Web Components. Viitattu 14.3.2021. https://developer.mozilla.org/en-US/docs/Web/Web_Components

Näpäri L. 2017. Haastattelun lajityypit. Viitattu 20.3.2021. <https://spoken.fi/2180/>

Orendorff J. 2015. ES6 In Depth: let and const. Viitattu 20.3.2021. <https://hacks.mozilla.org/2015/07/es6-in-depth-let-and-const/>

Pishchulin A. 2020. ServiceNow UI Framework. Viitattu 20.3.2021. <https://pishchulin.medium.com/servicenow-ui-framework-be88f466be01>

Pulkkanen A. Sinunkin kannattaa valita: 6 yleistä menetelmää projektityöhön (sis. Agile, Waterfall ja Kanban). Viitattu 29.1.2021. <https://www.agendum.com/post/agile-waterfall-kanban-6-projektinhallintamenetelmaa>.

Semaphore Partners, 2020. Leveraging Now Experience UI Components. Viitattu 29.1.2021. <https://semaphorepartners.com/posts/now-ui-components-orlando/>.

ServiceNow. 2021. Cloud Management. Viitattu 20.3.2021. <https://docs.servicenow.com/bundle/orlando-it-operations-management/page/product/cloud-management-v2/concept/cloud-management-v2-landing-page.html>

ServiceNow. 2021. INTRODUCTION TO THE NEW NOW EXPERIENCE UI BUILDER. Viitattu 20.3.2021. <https://developer.servicenow.com/blog.do?p=/post/quebec-intro-to-ui-builder/>

ServiceNow. 2020. NOW EXPERIENCE. Viitattu 20.3.2021. <https://developer.servicenow.com/blog.do?p=/post/now-experience/>

ServiceNow. 2021. Now® Experience UI Framework. Viitattu 14.3.2021. <https://developer.servicenow.com/dev.do#!/reference/now-experience/orlando/ui-framework/getting-started/introduction>

ServiceNow. 2021. Service Portal. Viitattu 20.3.2021. https://docs.servicenow.com/bundle/orlando-servicenow-platform/page/build/service-portal/concept/c_ServicePortal.html

Shopify. Benchmarking. Viitattu 2021. <https://www.shopify.com/encyclopedia/benchmarking>

Short T. 2020. What is SaaS? 10 FAQs About Software as a Service. Viitattu 20.3.2021. <https://www.softwareadvice.com/resources/saas-10-faqs-software-service/>

Tuominen K. Introducing Benchmarking. 2016. Viitattu 23.3.2021. <https://ebookcentral.proquest.com/lib/laurea/detail.action?docID=4883137>.

Wedekind K. 2018. 6 Reasons You Should Use Native Web Components. Viitattu 14.3.2021. <https://codeburst.io/6-reasons-you-should-use-native-web-components-b45e18e069c2>

White S, Greiner L. 2019. What is ITIL? Your guide to the IT Infrastructure Library. Viitattu 20.3.2021. <https://www.cio.com/article/2439501/infrastructure-it-infrastructure-library-til-definition-and-solutions.html>

Julkaisemattomat

Toimeksiantaja. 2020. Asiakashaastattelut.

Toimeksiantaja. 2020. SWOT-analyysi.

Taulukot

Taulukko 1: Käyttäjäprofiilit.....	11
Taulukko 2: ServiceNow järjestelmän SWOT-analyysi.....	16
Taulukko 3: Nykyisen järjestelmän SWOT-analyysi	17