

Testiympäristön Rakentamisen Automatisointi

Keinänen Kaius

Opinnäytetyö

Tietojenkäsittelyn koulutus
Tradenomi (AMK)

2024

Tekijä	Kaius Keinänen	Vuosi	2024
Ohjaaja(t)	Yrjö Koskenniemi		
Toimeksiantaja	Yritys X		
Työn nimi	Testiympäristön rakentamisen automatisointi		
Sivumäärä	28 + 2		

Opinnäytetyön tavoitteena oli suunnitella automaatio, joka pystyisi rakentamaan testiympäristöjä mahdollisimman yksinkertaisella automaatiolla. Opinnäytetyö suoritettiin suomalaiselle pelialanyritykselle, jolla on havaittu tarve testiympäristöjen määrän kasvattamisesta. Testiympäristöjen rakentaminen on työlästä ja niitä on tällä hetkellä liian vähän. Opinnäytetyössä selvitettiin, että miten toimeksianto voidaan toteuttaa, mitä ongelmia havaitaan ja mitä toimeksiantajan pitäisi seuraavaksi edistää.

Opinnäytetyössä käytettiin laadullista tutkimusmenetelmää ja laadullista tietoa kerättiin haastatteluiden avulla. Haastattelut toteutettiin videopuhelun välityksellä sekä sähköpostilla. Haastatteluista ilmeni, että tämän toimeksiannon toteuttamiselle on avoimia kysymyksiä liittyen toivottuun toteuttamiseen ja nykyiseen tilanteeseen. Tärkeimpänä nostona haastatteluista nousi, että työkalun käyttö pitäisi olla mahdollisimman yksinkertaista. Yksinkertaisuudella halutaan, että työkalua voisi käyttää ilman suurempaa teknistä osaamista.

Opinnäytetyön aikana suunniteltiin käyttötapausta, prosesseja ja myös arkkitehtuuria. Opinnäytteessä myös annettiin esimerkki mahdollisesta käyttöliittymästä, joka tukee yksinkertaista käyttöä. Opinnäytetyössä otettiin myös esille mahdollinen toteutus automaation osalta.

Toimeksiannon ongelmaksi muodostuu epätietoisuus nykytilanteesta. Testiympäristöjen rakentamiseen tarvitaan monen kehitystiimin työpanostusta. Jotta testiympäristön automatisoinnin työkalu olisi mahdollinen, pitää kehitystiimeillä olla omien komponenttien kohdalla myös automatisointia.

Opinnäytetyön avulla toimeksiantaja voi aloittaa työkalun toteuttamista. Toimeksiantaja näkee opinnäytetyön avulla, mitä tarvitaan työkalun toteuttamiseen ja mitä kannattaa huomioida. Ensimmäisenä toimeksiantajan suositellaan kartoittavan nykytilannetta yrityksessä, jotta saadaan paras kuva nykyisestä tilanteesta.

Avainsanat automaatio, ympäristö, ohjelmistotekniikka, testaus

Author	Kaius Keinänen	Year	2024
Supervisor(s)	Yrjö Koskenniemi		
Commissioned by	Organization X		
Title	Building a test environment with automation		
Number of pages	28 + 2		

The scope of this thesis was to provide a plan on how to develop an automation tool to build a test environment. The thesis was done for a Finnish gaming organization. The automation tool should be able to build a test environment from existing components. The organization is increasing their development, which means that in the future, the number of test environments must be increased. The building process of test environments takes time from multiple development teams. The thesis tries to answer the questions on how this automation tool can be done, what kinds of problems are found, and how the organization should continue in the future with this project.

This thesis was carried out by using qualitative research methods. Study was conducted through interviews via video calls and emails. The interviews provided information on what is required in the automation tool and what is the current state with automation. Simplicity of the automation tool was one of the most important aspects of the automation tool.

This thesis includes plans on use cases, processes and architecture in addition to providing an example on building a user interface which would be helpful when considering the wish for simplicity.

One of the greatest issues is the missing knowledge of automation usage within the organization. The test environment is built from existing components which are developed by separate development teams. These teams must have automation in use for this tool to work.

This thesis should give helpful information on how the automation tool could be done. The organization can utilize this thesis as a foundational guide for the automation tool project and as a roadmap for future actions.

Keywords automation, environment, software technology, testing

SISÄLLYS

1	JOHDANTO	5
1.1	Opinnäytetyön tavoite	5
1.2	Toimeksiantajan kehittämistarve	6
1.3	Tutkimusmenetelmät ja tiedonkeruu	8
2	OHELMISTOTESTAUS, AUTOMAATIO & PILVIPALVELUT	9
2.1	Ohjelmistotestaus	9
2.2	Automaatio ohjelmisto kehityksessä	9
2.3	Pilvipalvelut.....	10
3	HAASTATTELUJEN TOTEUTUS JA TULOKSET	11
4	TOIMEKSIANNON SUUNNITTELU.....	13
4.1	Projektin aloittaminen	13
4.2	Käyttötapaus.....	13
4.3	Prosessi.....	17
4.4	Arkkitehtuuri.....	18
4.5	Käyttöliittymä	19
4.6	CI/CD	23
4.7	Pilviratkaisu	24
5	POHDINTA	25
	LÄHTEET.....	27
	LIITTEET	29

1 JOHDANTO

Ohjelmiston testaaminen on ohjelmistojen kehityksen kannalta yksi tärkeimmistä jaksoista kehityksen aikana. Ohjelmistojen testaamisen aikana käydään läpi monenlaisia vaiheita, joissa tarkoituksena on taata tuotteelle haluttu laatu ja tuote toimii toivotusti. Testiympäristön valmistelu on yksi testaamisen tärkeimmistä vaiheista ja sen yleisesti toteuttaa erillinen tiimi. (Hooda & Chhillar 2015, 10.)

Testiympäristöjä käytetään testauksessa sen takia, että voidaan testata ohjelmiston laatua elävässä ympäristössä. Testausvaiheita voidaan suorittaa näissä testausympäristöissä ja ohjelmistoon voidaan testien perusteella tehdä korjauksia. Testiympäristön pitäisi olla mahdollisimman lähellä todellista tuotantoympäristöä, jotta laadun testaaminen olisi toimivaa. Testiympäristön käyttäminen testaamiseen on turvallinen tapa testata ohjelmiston laatua ennen tuotantoon vientiä. Testiympäristöjä voi myös olla erilaisia, joita käytetään eri testauksen vaiheissa. (Marget 2024.)

Automaatiota käytetään myös aktiivisesti testauksessa. Automaation kautta toteutettu ohjelmistojen rakentaminen on tehokas tapa pysyä mukana kehitystyössä esimerkiksi palvelun tai sovelluksen kehityksen aikana. Automaation avulla voidaan helposti ottaa mukaan halutut komponentit ja asentaa ne haluttuun ympäristöön. Manuaalinen ohjelmiston rakennus voi herkästi aiheuttaa virheitä. (Pathania 2016, 37.)

1.1 Opinnäytetyön tavoite

Tämän opinnäytetyön tavoitteena on suunnitella toimeksiantajalle ratkaisu, jota toimeksiantaja voi hyödyntää tämän projektin edistämiseen sekä antaa yritykselle mahdollisimman hyvä tilannekuva missä ollaan nyt ja mitä tarvitaan tämän projektin edistämiseksi. Opinnäytetyön tavoitteena ei ole tehdä lopullista teknistä toteutusta, vaan suunnitelma ratkaisusta. Jotta opinnäytetyöstä saadaan mahdollisimman hyödyllinen ja hyvä toimeksiantajalle, pitää selvittää toimeksiantajan tarpeet ja nykyinen tilanne. Opinnäytetyötä varten tietoa kerätään kirjallisuudesta sekä haastatteluista, joissa haastatellaan toimeksiantajan työntekijöitä.

Opinnäytetyön aihe on hyvä näyttö, mitä kaikkea opinnoista on mahdollista oppia, varsinkin kun keskitytään opinnäytetyössä uuden it-palvelun suunnitteluun. Opinnoissa käsitellään ja harjoitellaan paljon projektien edistämistä ja it-palveluiden suunnittelua sekä niiden kehittämistä.

Opinnäytetyössä pyritään vastaamaan seuraaviin kysymyksiin:

- Kuinka voidaan toteuttaa automatisoitu testiympäristön luonti ottaen huomioon muut sidosryhmät ja komponentit?
- Mitä haasteita tämä toimeksianto tuottaa toimeksiantajalle?
- Kuinka toimeksiantaja voi edistää tätä projektia?

Näiden kysymysten avulla pyritään saavuttamaan toimeksiannon tavoitteet. Opinnäytetyön tavoitteet koetaan onnistuneeksi, jos vastaukset edellä mainittuihin kysymyksiin selviävät. Näiden kysymysten pohjalta saadut vastaukset voisivat näyttää toimeksiantajalle suuntaa mahdollisista toimenpiteistä, ja tämän lisäksi toimeksiantaja hyötyy vastauksista projektin kannalta.

1.2 Toimeksiantajan kehittämistarve

Toimintaympäristönä toimii suomalainen pelialan yritys, josta käytetään opinnäytetyössä nimeä "Yritys x". Yritys x tuottaa erilaisia pelejä monenlaisiin asiakaskohderyhmiin ja on teknologiayrityksenä kasvavassa tilanteessa. Opinnäytetyössä keskitytään Yritys x:n antamaan toimeksiantoon, jonka tarkoituksena on suunnitella testiympäristöjen rakentamisen automaatiotyökalu. Toimeksiantajan ongelmaksi on muodostunut testiympäristöjen määrä. Toimeksiantajan mukaan eri tuotteiden samanaikainen kehitys aiheuttaa ruuhkaa testiympäristöjen käyttöön, joten testiympäristöjä ei ole riittävästi käytettäviksi kaikille tahoille yrityksen sisällä. Ongelmaksi muodostuu myös testiympäristöjen luonti, sillä näiden luominen vaatii aina aikaa sekä jonkun tahon työpanosta. Tämä johtaa aikatauluongelmiin, kun testiympäristön luominen ei aina onnistu heti. Saamani toimeksianto on siis suunnitella automaatio, joka mahdollistaisi testiympäristön luonnin napinpainalluksella.

Yritys x käyttää testiympäristöjä tuotteidensa ja sisäisten järjestelmiensä kehittämiseen ja testaamiseen. Yritys x kehittää erilaisia pelejä, joita pelataan joko mobiiliapplikaation kautta tai verkkoselaimen kautta. Yrityksessä käytettävät testiympäristöt näyttävät yleisesti tuotannon kaltaiselta sivustolta, mutta niistä saattaa puuttua paljon ominaisuuksia eikä niissä käytetä oikeita asiakastietoja. Kyseessä on melko suuri yritys, jonka tarkoituksena on kasvaa kansainvälisillä markkinoilla ja tämän myötä tuotteiden määrä on kasvamassa, joten myös sisäisiä järjestelmiä on tarkoitus kehittää entisestään. Tiettyjä tuotteita voidaan testata samassa ympäristössä ja sitä tehdäänkin paljon. Yrityksellä on jo olemassa monia testiympäristöjä, joista kaksi todella suurta testiympäristöä, joiden tarkoituksena on simuloida mahdollisimman tarkasti tuotannon kaltaista tilannetta.

Toimeksiantajan toiveena on luoda palvelu yrityksen sisäiseen käyttöön, joka mahdollistaisi helpomman testiympäristöjen hallinnan ja luonnin. Palvelun avulla testiympäristön luonti voidaan suorittaa painamalla palvelussa olevaa nappia, joka luo testiympäristön automaattisesti. Käyttäjä, joka haluaa luoda testiympäristön voi määrittää, mitä komponentteja kyseiseen testiympäristöön halutaan tuoda. Tämän toimeksiannon yhteydessä on annettu toive kehittää samalla ominaisuus, jolla testiympäristö olisi helppo tuhota. Eli käytännössä testiympäristön tuhoaminen olisi yhtä yksinkertaista, kuin testiympäristön luonti. Testiympäristöjen luontia ja tuhoamista varten olisi tarkoitus myös kehittää joku sivusto, mistä tämän toiminnon voisi suorittaa.

Testiympäristöjen rakentamisen automatisoinnilla halutaan varmistaa, että yksittäiset kehitysprojektit saavat oman kehitysympäristön, tässä tapauksessa testiympäristön. Tällöin samassa ympäristössä ei olisi yhtäaikaaisesti muita tahoja testaamassa omia palveluita tai komponentteja, vaan projektia voisi rauhassa kehittää ja testata omassa ympäristössään. Kun kyseinen palvelu tai komponentti on kehitetty riittävän pitkälle, voidaan se siirtää aikaisemmin mainittuun suurempaan testiympäristöön, jossa voidaan toteuttaa integraatiotestausta - eli testausta, jossa katsotaan, että komponentit ja palvelut toimivat yhteen (ISTQB 2023).

Uuden testiympäristön luojalla pitäisi olla mahdollisuus itse määrittää oman tarpeensa mukaan mitä testiympäristö pitää sisällään, mutta joitain asioita on kuitenkin pakko sisällyttää, kuten esimerkiksi frontend-käyttöliittymä. Eli testiympäristö luodaan aina verkkosivupohjaiseksi, jolloin sivusto saattaa näyttää tuotannon kaltaiselta sivustolta, mutta sieltä voi puuttua tärkeitä ominaisuuksia riippuen siitä, mitä testiympäristöön on haluttu.

1.3 Tutkimusmenetelmät ja tiedonkeruu

Tässä opinnäytetyössä tullaan käyttämään laadullista tutkimusmenetelmää. Laadullinen tutkimus, eli kvalitatiivinen tutkimus on menetelmä, jonka avulla pyritään ymmärtämään kohteen ominaisuuksia, merkitystä ja laatua kokonaisvaltaisesti. Laadullista tutkimusta voidaan toteuttaa erilaisin menetelmin (Jyväskylän Yliopisto, 2021.) Laadullinen tutkimus menetelmänä sopii hyvin tähän toimeksiantoon, sillä tästä aiheesta löytyy suhteellisen vähän tietoa tällä hetkellä. Surveymonkey sivustolla kerrotaan näin: ”Laadullisella tutkimuksella voidaan selvittää tutkimuslöydösten syitä. Se keskittyy vähemmän tilastoihin ja jäseneltyyn dataan ja auttaa ymmärtämään syvällisesti ihmisten käyttäytymistä, motiiveja ja tunteita tekstimuotoisen tiedon kautta.” (Surveymonkey 2024.) Kvalitatiivinen tutkimus sisältää paljon erilaisia menetelmiä tiedonkeräämiseen ja on sen myötä soveltuva tähän opinnäytetyöhön. Se tarkoittaa sitä, että kvalitatiivinen tutkimus ei ole vain tietyn tieteenalalan tutkimusote tai vain yksi tapa tutkia. (Saaranen-Kauppinen & Puusniekka 2006).

Opinnäytetyössä tarvitaan aineistoa, joka liittyy jossain määrin testaukseen ja ympäristöjen luontiin. Opinnäytetyöhön tarvitaan myös kirjallisuutta projektien läpiviemiseen ja prosessien luontiin. Juuri tämän toimeksianton tehtävään ei välttämättä löydy suoraan kirjallisuutta, joka osaisi kertoa tarkemmin aiheesta. Tämän takia, kun käytetään laadullista tutkimusmenetelmää ja hyödynnetään mahdollisimman paljon jo olemassa olevaa kirjallisuutta, voidaan mahdollisesti saada kokonaiskuva tarpeista ja haasteista.

Aineiston hankinta tehdään haastattelujen avulla, jotta saadaan paras mahdollinen tieto suoraan toimeksiantajalta. Haastattelujen kautta voidaan saada tärkeää tietoa ja toiveita toimeksiantannon läpiviemiseen.

2 OHELMISTOTESTAUS, AUTOMAATIO & PILVIPALVELUT

2.1 Ohjelmistotestaus

Ohjelmistotestaamisella tarkoitetaan tapaa testata jotain tiettyä komponenttia tai ohjelmistoa. Testaamisen tarkoituksena on verifioida, että ohjelma toimii halutulla tavalla ja että, ohjelma vastaa sitä mitä on tilattu. Testaamista voidaan suorittaa erilaisten aktiviteettien kanssa, minkä kautta löydetään ongelmia tai virheitä ohjelmasta. Testaaminen ei ole pelkästään ohjelmien nappien painamista tai pelin pelaamista, vaan testaamisen suunnitteluun käytetään aikaa ja testituloksia analysoidaan. (International Software Testing Qualifications 2023, 15.)

Toimeksianto on suunnattu Yritys x:n laadunvarmistukselle, joka hoitaa yrityksessä ohjelmisto- ja pelitestauksen. Yrityksen laadunvarmistuksella on ollut jo pitkään tarve saada tämän tyyppinen automaatio työkalu. Yrityksessä testataan ohjelmia, komponentteja ja pelejä, jotka vaativat testiympäristöjä toimiakseen. Testaaminen suoritetaan yrityksessä monella tapaa ja tuotteita on todella paljon.

Tämän toimeksiannon projekti tulee keskittymään enimmäkseen testaukseen mitä suoritetaan pääsääntöisesti selain pohjaisesti, tietokoneilla ja mobiililaitteilla. Uusien tuotteiden valmistuttua niitä ei viedä suoraan asiakkaan käytettäväksi, vaan ne käyvät läpi tarkoin määritetyn testausprosessin. Kehityksen ja testauksen aikana käytetään testiympäristöjä.

2.2 Automaatio ohjelmisto kehityksessä

Jatkuva julkaiseminen (continuous integration) on todella hyödyllinen automaatio ohjelmistokehitykselle. Jatkuva julkaiseminen tunnetaan myös lyhenteellä CI/CD. Tämän tavoitteena on antaa kehittäjille mahdollisuus julkaista koodimuutoksia paljon useammin. Jatkuvan julkaisemisen idea mahdollistaa pienien koodimuutosten viennin. Moni ohjelmakehitys vaatii käytettäväksi erilaisia alustoja ja työkaluja, joten kehittäjät tarvitsevat tavan tuoda koodia nopeammin käyttöön ja verifioitavaksi. Jatkuva julkaiseminen on automaatio, joka rakentaa ohjelman tiettyyn ympäristöön esimerkiksi testiympäristöön. Jatkuva

julkaiseminen käytännössä toimii niin, että CI/CD työkalu säilyttää ympäristöön vaikuttavat parametrit, joita tarvitaan pakettien viemiseen ympäristöön. Työkalu osaa tehdä kaikki tarvittavat kutsut tiettyihin servereihin ja tietokantoihin. (Sacolick, 2022.)

Opinnäytetyössä on tarkoitus selvittää, kuinka hyvin voidaan hyödyntää CI/CD:n kaltaista työkalua ja onko tähän opinnäytetyöhön jotain puutteita tämän työkalun suhteen. Yritys X:ssä käytetään näitä mainittuja automaatioita jo, mutta ei tähän kyseiseen opinnäytetyön aiheeseen.

2.3 Pilvipalvelut

Pilvipalvelu on palvelu joka mahdollistaa käyttäjän pääsyn palveluihin internetin kautta, kuten esimerkiksi servereihin, tietokantoihin tai ohjelmaan paikasta riippumatta. Tietoa tai ohjelmia ei tarvitse säilyttää omalla tai yrityksen kovalevyllä tai serverillä, vaan niitä voidaan säilyttää etänä muualla. Internetin välityksellä yhteyden teko antaa mahdollisuuden käyttää tietoja tai ohjelmaa siellä, missä internetyhteys on. Pilvipalveluista on tullut todella suosittuja yksityishenkilöille ja yrityksille, koska se on kustannus tehokasta, nopeaa ja turvallista. (The Investopedia Team 2024.)

Opinnäytetyössä pitää ottaa huomioon myös pilvipalvelut. Testiympäristö pitää rakentaa jonkin ja tässä tapauksessa, se on pilveen. Pilvipalveluiden käyttö laajenee ja myös Yritys x:ssä on siirretty paljon komponentteja ja palveluita pyörimään pilvialustalle. Testiympäristö rakennetaan pilveen, koska Yritys x:llä ei ole käytössä omia palvelimia. Kaikilla tarvittavilla henkilöillä on pääsy ympäristöön, joka asuu pilvipalvelussa. Opinnäytetyössä pitää selvittää, mitä kaikkea se vaatii, että saadaan testiympäristön asennettua pilveen automaation avulla.

3 HAASTATTELUJEN TOTEUTUS JA TULOKSET

Haastatteluja tein kolmelle työntekijälle, jotka ovat aktiivisesti mukana joko laadunvarmistuksen kanssa tai tietävät miten tätä toimeksiantoa kannattaisi lähteä toteuttamaan. Haastatteluista yksi tehtiin videopuhelun avulla ja kaksi muuta haastattelua tehtiin sähköpostin välityksellä. Haastattelukysymykset ovat esiteltyinä tämän opinnäytetyön liitteessä 1. Pyrin miettimään haastattelukysymykset mahdollisimman suoraviivaiseksi, jotta saataisiin hyvää ja helposti käytettävää tietoa toiveista ja tarpeista. Haastattelut sujuivat hyvin ja tarpeellista tietoa saatiin haastatteluista, joita pystytään hyödyntämään toimeksiannon suunnittelussa.

Testiympäristöjen tärkeys korostui haastatteluissa. Testiympäristöjä käytetään Yritys x:ssä paljon ja niissä testataan tuotteiden laatua, ennen kuin ne viedään tuotantoon. Testiympäristöjen toiminta on suoraan kytköksissä liiketoiminnan kehityshankkeisiin. Testiympäristöistä myös nostettiin esille, että tärkeänä pidetään sitä, että testiympäristöt vastaavat tuotannonkaltaista ympäristöä, jotta ohjelmien ja palveluiden toimivuuden varmistaminen on toimivaa. Haastateltavat itse eivät välttämättä ole suoraan aina tekemisissä testiympäristöjen kanssa. Kaikilla haastateltavilla oli kuitenkin suuri kiinnostus testiympäristöjä kohtaan, koska testiympäristöjen toimivuus vaikuttaa moneen asiaan, esimerkiksi laadunvarmistuksessa.

Tämän toimeksiannon hyödyllisyydestä löytyi yksimielisesti yksi todella suuri etu. Tämä etu liittyy siihen miten nykyään yrityksessä rakennetaan testiympäristöjä. Testiympäristöjen rakentaminen vaatii monen kehitystiimin työpanostusta, jonka myötä helposti tulee paljon työtunteja ympäristön luomiseen. Tämä myös nopeasti vaikuttaa testiympäristöjen luonnin nopeuteen, sillä kehitystiimeillä voi olla samanaikaisesti paljon muutakin tekemistä. Toimeksiannon työkalun avulla voitaisiin laadunvarmistuksen sisällä luoda testiympäristöjä ja tämän myötä vapauttaa resursseja kehitystiimeiltä. Huolena nousi, että tulevaisuudessa testiympäristöjen tarve tulee näillä näkyvin kasvamaan, joten nopeampi ja tehokkaampi tapa testiympäristöjen luontiin olisi todella hyödyllistä.

Teknisesti tämän toimeksiannon läpivieminen haastattelujen perusteella vaatii paljon nykyisten prosessien tutkimista ja kuinka paljon automatisointia on jo olemassa, mitä voidaan hyödyntää tähän toimeksiantoon. CI/CD:n putkea on mahdollisuus käyttää, mutta se tarkentuu vasta, kun tiedetään missä tällä hetkellä mennään muun automatisoinnin osalla esimerkiksi muiden kehitystiimien näkökulmasta katsoen. Haastatteluissa nousee esille, että ei kannata lähteä liian suurella paineella liikkeelle tämän työkalun kehittämisen kanssa, vaan kannattaisi suunnitella hyvin ja lähteä pienistä asioista liikkeelle. Myös vanha ajattelutapa pitää kuulemma unohtaa. On tärkeää, että kaikki ketkä osallistuvat tämän automaation rakentamiseen, ovat avoimia uusille ideoille.

Käyttöliittymä on toivottua. Ilman käyttöliittymää on mahdollista elää, mutta suunnitteluun olisi hyvä miettiä käyttöliittymää myös. Haastatteluissa nousee esille sen yksinkertaisuus, eli tarkoituksena on ,että se olisi helppokäyttöinen ja sitä voisi nopeasti oppia käyttämään, ilman suurempaa teknistä osaamista. Haastatteluista jäi päällimmäisenä mieleen, että yksinkertaisuus on yksi tämän toimeksiannon suurista toiveista.

4 TOIMEKSIANNON SUUNNITTELU

4.1 Projektin aloittaminen

Tämän kokoluokan automaation kehittämiseen tarvitaan projekti muotoinen toteutustapa. Haastattelujen perusteella tähän automaatioon kuuluu melko suuri määrä työntekijöitä, koska se vaikuttaa moneen kehitystiimiin. Kehitystiimit pitää saada mukaan tähän projektiin, jotta se olisi toimivaa, sillä jos tietyn kehitystiimin palvelu tai komponentti ei ole automatisoitu, ei automaatiolla voida luoda testiympäristöä. Projektin kokonaisuuteen ei periaatteessa siinä tapauksessa kuulu vain automaation työkalun tekeminen, vaan myös kehitystiimien pitää käyttää automaatiota.

Projekti aloitetaan aina sillä, että voidaan vastata seuraaviin kysymyksiin:

- Mitä me teemme?
- Ja mistä me tiedämme, että me teimme sen?

Näiden kysymysten avulla on tarkoitus ymmärtää projektissa, mitä tehdään ja kuinka tiedetään, että tavoite on saavutettu. (Wysocki 2019, 153–154.)

Projekti täytyy suunnitella hyvin, joten tämä opinnäytetyö keskittyy siihen. Hyvin suunniteltu projekti antaa suuremman mahdollisuuden onnistumiseen, verraten huonosti suunniteltuun projektiin. Hyvä suunnittelu vähentää epävarmuutta projektin aikana, se parantaa projektin ymmärrettävyyttä ja mahdollistaa myös tehokkaamman työskentelyn. (Wysocki 2019, 192–194.)

Toimeksiannon myötä, tiedetään mitä halutaan tehdä. Opinnäytetyössä suunnitellaan työkalua, joka rakentaa testiympäristön. Projekti voidaan todeta valmiiksi, kun on olemassa toimiva automaatio, joka pystyy rakentamaan testiympäristön.

4.2 Käyttötapaus

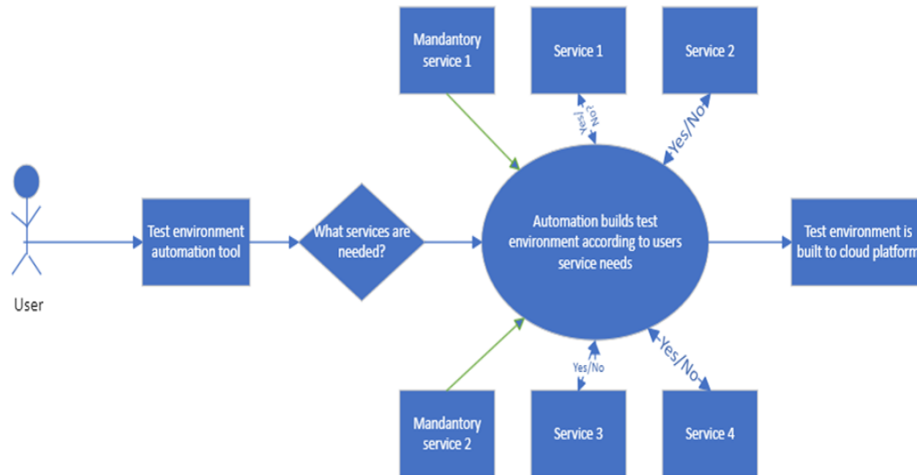
Ensimmäisenä täytyy selvittää mitä käyttötappauksia automaatiotyökalulla on. Toimeksiantajan pyynnön perusteella itse käyttötapaus on melko yksinkertainen.

Tarvitaan palvelu, joka pystyy luomaan testiympäristön yhden napin painalluksella. Lähdetään ratkomaan käyttötapausta.

Käytetään käyttötapauskaaviota (kuvio 1) helpottamaan käyttötapausten esittelyä. Käyttötapauskaaviolla voidaan esittää käyttäjien kautta, miten ohjelman pitäisi toimia ja tarjota. Moni kokee käyttötapauskaavion hyödyllisenä itselleen. Käyttötapauskaaviossa on käyttäjä (actor), joka esittää roolia, miten jokin käyttäjä on yhteydessä ohjelmaan tai tässä tapauksessa palveluun. Käyttäjän avulla voidaan esittää mitä esimerkiksi käyttäjä tekee ohjelmassa tai palvelussa. (Fowler & Scott 1999, 51–52)

Kuva 1 ei välttämättä ole perinteinen käyttötapauskaavio mutta se antaa mahdollisuuden näyttää mitä tehdään. Tilanteessa on siis käyttäjä, joka haluaa luoda testiympäristön. Käyttäjälle pitää ensin luoda jokin rajapinta mistä käyttäjä pääsee tekemään haluamansa valinnat testiympäristöä varten. Rajapinnaksi voisi kehittää erillisen html sivuston. Tämän sivuston kautta käyttäjä pääsee kertomaan palvelulle mitä tarvitaan testiympäristöön. Sen jälkeen, kun palvelut ja komponentit on valittu testiympäristöön voi rakentaminen alkaa. Kuten kaaviossa näkyy ovat jotkin palvelut pakollisia. Yritys x:n testiympäristöissä on joitakin pakollisia palveluita, jotta testaaminen on mahdollista kyseisessä testiympäristössä. Esimerkki pakollisesta komponentista on frontend-toteutus eli itse visuaalinen verkkosivu toteutus, jossa testaaminen suoritetaan. Kuva 1:sen keskellä oleva

iso ympyrä kuvaa ympäristön rakentamista. Sen jälkeen ympäristö pitäisi rakentaa jollekin alustalle. Yritys x ei käytä omia palvelimia, joten tarvitaan pilvialusta, jonne rakennetaan testiympäristö.



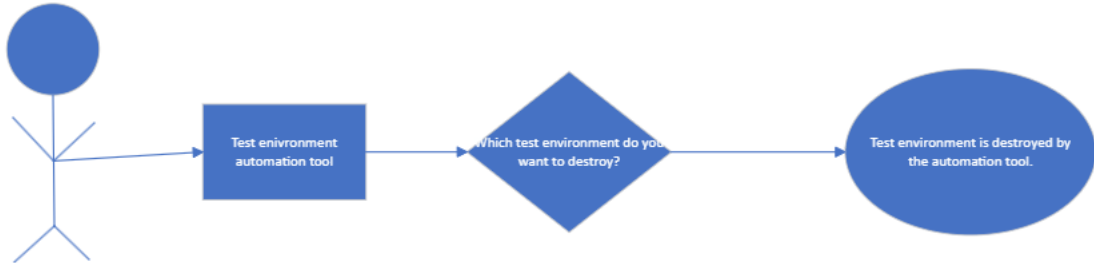
Kuva 1. Käyttötapauskaavio

Kaaviosta nähdään, että käyttäjän näkökulmasta tämä rakentamisen prosessi on todella yksinkertainen ja helppokäyttöinen. Käyttäjällä on vain valinnat, että mitä testiympäristöön halutaan ja sen jälkeen voidaan rakentaa testiympäristö automaatiolla. Yksinkertaisuus on tämän toimeksiannon yksi tärkeimmistä elementeistä. Suunnitellun palvelun pitäisi olla niin helppokäyttöinen, että myös vähemmän tekninen työntekijä pystyisi ja osaisi rakentaa kyseisellä automaatio palvelulla testiympäristön.

Vaikka minun tekemä käyttötapauskaavio poikkeaa hieman niin sanotusta tavanomaisesta käyttötapauskaaviosta, nähdään kaavioista se mitä tahdotaan. Käyttötapauskaavio näyttää korkealla tasolla ohjelman toiminnot ja tavoitteen. Kaavio myös näyttää käyttäjän ja ohjelman yhteyden ja näyttää niiden välisen kanssakäymisen. (IBM 2021.)

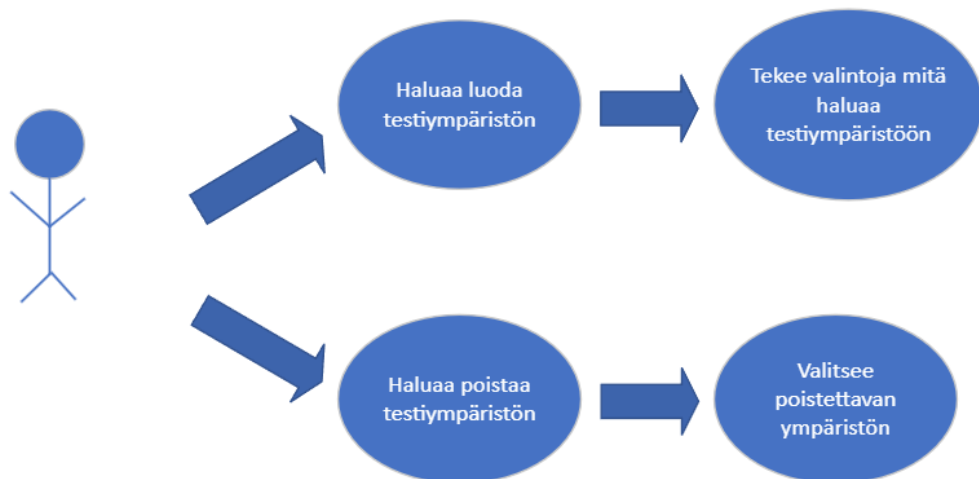
Käyttötapauskaavio pitää myös tehdä testiympäristön tuhoamisesta. Kuva 2 on melko samanlainen kuin kuva 1, koska käyttäjän tapa toimia on melkein sama. Käyttäjä menee palveluun, valitsee poistettavan testiympäristön ja hyväksyy poistamisen. Tämän jälkeen automaatio työkalu tuhoaa halutun ympäristön ja

kaikki siihen kuuluvat komponentit ja palvelut mitä siihen on ollut asennettuna. Testiympäristö poistetaan pilvestä ja näin voidaan säästää kustannuksia, kun käyttämätön testiympäristö on otettu pois käytöstä pilvipalvelusta.



Kuva 2. Käyttötapauskaavio tuhoamisesta

Kuvassa 3 näytetään molemmat käyttötapaukset samassa mutta vielä korkeammalla tasolla. Eli käyttäjä voi luoda testiympäristön valitsemalla haluamansa palvelut ja komponentit, jonka jälkeen testiympäristö luodaan pilvipalveluun. Käyttäjä voi myös poistaa testiympäristön. Hän valitsee minkä testiympäristön haluaa poistaa ja vahvistaa poiston.



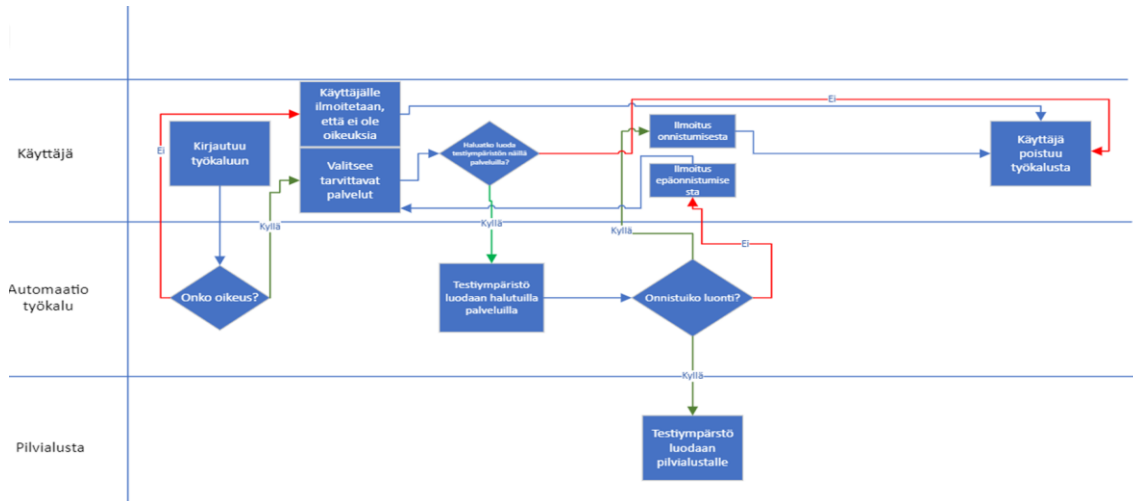
Kuva 3. Käyttötapauskaavio yksinkertaistettuna

4.3 Prosessi

Tämän palvelun prosesseja lähdetään suunnittelemaan ehkä hieman vanhalla tavalla, mutta vieläkin toimivalla tavalla. Otetaan esille vanhasta UML (Unified Modeling Language) oppaasta toimintakaavio (activity diagram). Toimintakaavio avulla voidaan kuvata aktiviteetteja ja mitä tietyn aktiviteetin jälkeen tapahtuu. Toimintakaavio avulla voidaan siis kuvata käytännössä esimerkiksi ohjelman prosessi. Voidaan kuvata mitä tapahtuu, kun käyttäjä painaa jotain nappia. Toimintakaaviossa voidaan myös käyttää uimaratoja (swimlanes) joiden avulla voidaan kuvata kaavioon kuka tai mikä tekee jotain. Se selkeyttää kaaviota, koska se näyttää samalla toimintaa ja aktiviteetteja mutta uimaratojen avulla voidaan helposti visualisoida kuka tai mikä tekee tietyn aktiviteetin. (Fowler & Scott 1999, 147–157)

Kuvassa 4 nähdään tähän toimeksiantoon suunniteltu toimintakaavio, jossa hyödynnetään uimaratoja. Oletetaan, että työkalun käyttö on suojattu käyttöoikeuksilla, koska vaikka työkalun tarkoituksena on tehdä helposti käytettävä työkalu, jota kuka tahansa voisi käyttää se ei tarkoita sitä, että kaikilla on oikeus sitä käyttää. Tähän tarpeeseen tarvitaan jokin käyttöoikeus. Kaaviossa lähdetään liikkeelle käyttäjän uimaradasta, jossa käyttäjä kirjautuu työkaluun. Automaatio työkalu tarkistaa onko käyttäjällä oikeutta käyttää tätä työkalua ja jos on, pääsee käyttäjä eteenpäin työkalussa. Käyttäjä saa valita tarvittavat palvelut mitä testiympäristöön halutaan luoda.

Kun valinnat ovat tehty pyydetään vahvistus käyttäjältä, jossa kysytään, haluatko luoda ympäristön näillä valinnoilla. Tässä vaiheessa on vielä mahdollisuus kieltäytyä ja testiympäristöä ei luoda. Jos käyttäjä haluaa luoda työkalun näillä palveluilla, työkalu aloittaa ympäristön luonnin. Työkalua ilmoittaa ympäristön luonnin jälkeen onko luonti onnistunut vai ei. Jos se on onnistunut, ympäristö on pilvipalvelussa. Jos jostain syystä ympäristön luonti ei onnistunut, siitä tulee käyttäjälle ilmoitus, että ympäristön luonti epäonnistui. Tämä toimintakaavio ei välttämättä kerro kaikkea mitä tapahtuu työkalun kanssa, mutta tämän kaavio avulla nähdään mitä tapahtuu, kun käyttäjä käyttää työkalua.



Kuva 4. Toimintakaavio

4.4 Arkkitehtuuri

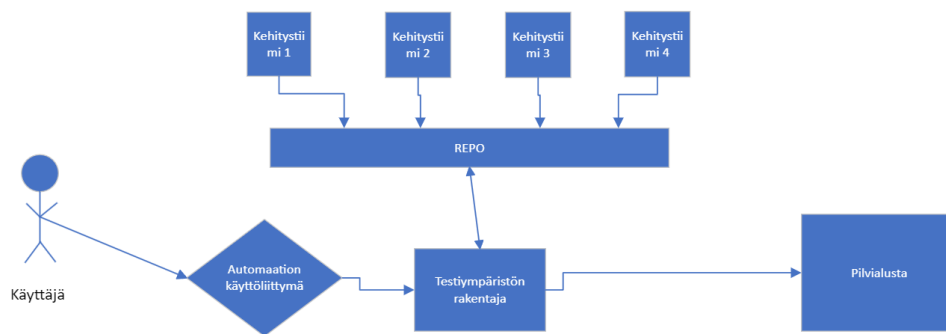
Arkkitehtuuri ohjelmisto kehityksessä tarkoittaa yleisesti kokonaiskuvan hahmottamista. Arkkitehtuurin avulla voidaan kuvata ohjelman rakennetta, joka sisältää komponentteja, rajoituksia ja toimintoja. Toimintoja kuvataan arkkitehtuuri piirustuksessa, sillä tavoin, että voidaan nähdä miten eri komponentit toimivat toisten komponenttien kanssa. Arkkitehtuuria voidaan kuvata monesta eri näkökulmasta. Arkkitehtuuriin voidaan kuvata tietokoneen komponentteja, ohjelmia ja verkkoa. Arkkitehtuurin hyvä suunnittelu tuo helpotusta esimerkiksi ohjelman käytettävyyteen, joten arkkitehtuurin vaikutukset voivat vaikuttaa moneen eri osa-alueeseen. (Hoffman, Grivel & Battle 2005.)

Todetaan, että tämän toimeksiannon toteuttamiseksi, on hyvä suunnitella arkkitehtuuri hyvin. Ongelmana tässä on, että en tunne Yritys x:n muuta ympäröivää arkkitehtuuria riittävän hyvin, jotta voitaisiin luoda paras mahdollinen arkkitehtuuri tälle työkalulle. Haastattelujen perusteella voidaan kuitenkin todeta asioita ja saada näkemystä yrityksen arkkitehtuurista ja mitä opinnäytetyössä pitää huomioida.

Haastatteluissa ilmenee, että testiympäristöjen rakentamiseen osallistuu moni kehitystiimi. Kehitystiimit Yritys x:ssä ovat vastuussa omista kehitysprojekteistaan. Jos testiympäristöön halutaan komponentti tai palvelu käyttöön, tarvitsee

se työtä sen palvelun tai komponentin omistavalta kehitystiimiltä. Arkkitehtuuri kuvauksessa pitää huomioida kyseisiä kehitystiimejä ja niiden omistamia palveluita tai komponentteja. Haastatteluista käy myös ilmi, että työkalun pitäisi olla mahdollisimman yksinkertainen, joten jos viitataan artikkeliin ”Designing software architectures to facilitate accessible web applications”, nähdään että tässä vaiheessa hyvin suunniteltu arkkitehtuuri voi samalla tuoda yksinkertaisuutta työkaluun. (Hoffman, Grivel, Battle 2005.)

Luodaan todella yksinkertainen arkkitehtuuri kuvaus, joka ei vastaa yrityksen todellista arkkitehtuuria, mutta tämän kautta voidaan esittää suunnitelma arkkitehtuurista. Ajatuksena tässä arkkitehtuurissa on, että jokainen kehitystiimi työstää omaa komponenttiaan ja tekee uusia versioita kuvassa 5 näkyvään ”repoon”, eli paikkaan missä koodia säilytetään. Kaikilla kehitystiimeillä on luultavasti oma sijainti, missä koodia säilytetään. Käyttäjä pääsee hallinnoimaan käyttöliittymän kautta, mitä testiympäristöön halutaan ja itse testiympäristön rakennusautomaatio hakee näistä säilytyspaikoista koodit, joita kehitystiimit ovat luoneet. Automaatio rakentaa nämä kaikki komponentit pilvialustalle.



Kuva 5. Arkkitehtuuri

4.5 Käyttöliittymä

Haastattelujen perusteella käy ilmi, että toiveena olisi saada käyttöliittymä, jonka kautta testiympäristöjen rakentamisen automatisoinnin työkalua pystyisi käyttämään. Haastatteluissa mainittiin, että se ei välttämättä olisi alkuun tarpeellista, mutta se olisi todella helpottava tekijä. Haastatteluissa nousi myös esille, että

työkalun sekä testiympäristöjen rakentamisen automatisoinnin työkalu pitäisi olla helppokäyttöinen. Käyttöliittymän suunnitteluun yleisestikin liittyy yksinkertaisuus. Käyttöliittymä pitäisi tehdä käytännössä sen tasoisesti helpon, että siinä ei tarvitse miettiä mitä painaa tai tekee. Käyttäjän pitäisi ymmärtää käyttöliittymästä, mitä hän voi tehdä ja millä tavalla hän voi tehdä näin. Steve Krug (2014, 29) painuu juuri tähän aiheeseen. Hän kiinnittää huomiota siihen tosiasiaan, että kun käyttäjä tulee käyttöliittymään, ei hänen tarvitsisi ajatella sen enempää käyttöliittymän toimivuutta, vaan se toimii niin helposti, että ei käytännössä tarvitse sen enempää ajatella.

Kun suunnitellaan opinnäytetyön toimeksiannon automaatiota, voidaan todeta myös, että sen käyttämisen pitäisi olla mahdollisimman helppoa. Käyttöliittymän pitää olla myös selkeä, jotta ei tapahtuisi vahinkopainalluksia. Lähdin suunnittelemaan todella yksinkertaista käyttöliittymäideaa tämän työkalun käyttöä varten. Käyttöliittymä on tehty html:ää, css:ää ja Javascriptiä hyödyntäen. Kyseessä ei ole kompleksi kokonaisuus ja esimerkiksi napit eivät tässä html kokonaisuudessa toimi, mutta tämän kautta nähdään, mitä esimerkiksi työkalun käyttöliittymä voisi pitää sisällään.

Ensimmäisenä kehitettiin yksinkertainen kirjautumissivu (kuva 6). Kirjautumissivulla on vain otsikko, jonka alla on kirjautuminen itsessään. Sivun ei välttämättä tarvitse muuta, paitsi jos toimeksiantajalla on tarvetta jollekin muulle. Itse kirjautumiseen en ota kantaa. En tiedä, millä tunnuksilla Yritys x:n työntekijät haluavat

kirjautua työkaluun. Tämän takia html koodissa, ei ole kirjoitettu itse kirjautumista, eli sivu ei toimi. Kyseessä on vain esittely visuaalisesta esimerkistä.

Kirjaudu sisään



Käyttäjä:

Salasana:

Kuva 6. Kirjautuminen

Seuraavaksi tarvitaan itse sivu, missä valinnat ympäristön rakentamiseen tehdään (kuva 7). Tässä on kyse tämän kokonaisuuden tärkeimmästä sivusta, koska täällä tehdään valinnat mitkä palvelut/komponentit valitaan kyseiseen testiympäristöön mikä rakennetaan. Tähän sivustoon toimeksiantajalla voi olla toiveena

saada paljon enemmänkin vaihtoehtoja tai tekstiä, mutta saatujen tietojen perusteella ei voida lisätä siihen enempää. Sivun suunnittelussa taas käytettiin mahdollisimman yksinkertaista lähestymistapaa.

Testiympäristön rakennus

Anna nimi testiympäristölle

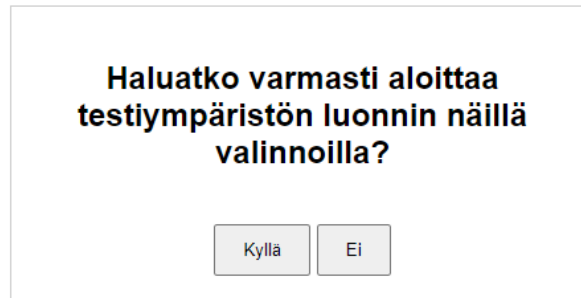
Valitse haluamasi palvelut testiympäristöön

- Service 1
- Service 2
- Service 3
- Service 4

Kuva 7. Testiympäristön rakentaminen

Kun käyttäjä on valinnut haluamansa komponentit/palvelut, voi hän painaa ”rakenna” painiketta. Rakenna-painike aloittaa itse testiympäristön rakentamisen. Ennen rakentamisen aloittamista, on kuitenkin hyvä varmistaa käyttäjältä, että ollaanko varmoja, että testiympäristö halutaan rakentaa näillä palveluilla (kuva 8). Vahvistaminen on hyvä tapa vahvistaa käyttäjältä, ennen kuin tehdään jotain millä voi olla jotain negatiivisia vaikutuksia (Aitman, 2016). Tässä tilanteessa negatiivinen vaikutus on turhan ympäristön rakentaminen. Turha ympäristö voi tuot-

taa nopeasti turhia kustannuskuluja yritykselle (LaunchDarkly 2021). Seuraavassa kuvassa näytetään ehdotus ikkunasta, jossa on kysymys, että halutaanko varmasti aloittaa testiympäristön rakentaminen.



Kuva 8. Vahvistus

Näiden sivujen yksinkertaisuuden kautta voidaan päästä tilanteeseen, missä käyttäjän ei tarvitse miettiä. Sivun yksinkertaisuudella ja helposti luettavuudella halutaan taata se, että käyttäjän ei tarvitse miettiä, että miksi on täällä, mitä minun pitäisi tehdä tai mikä tämä sivu on (Krug 2014, 35).

4.6 CI/CD

Automaatiotyökalun tekninen puoli voi olla vaikeampi määrittää, kun ei tunneta toimeksiantajan kokonaisarkkitehtuuria. Tiedetään kuitenkin, että kyseessä on monen kehitystiimin työpanos, kun puhutaan testiympäristön rakentamisesta. On hyvä kartoittaa, voiko työkalun toimivuutta toteuttaa hyödyntäen CI/CD putkia. Voidaan tutkia CI/CD "Multi-Repo" lähestymistapaa. "Multi-Repo:lla" tarkoitetaan struktuuria, joka pitää sisällään eri projekteja tai komponentteja omissa "säiliöissä". Tämä antaa mahdollisuuden, että kehitystiimit voivat työskennellä omilla halutuilla tavoillaan ja tehdä omia muokkauksia komponenttiin ilman, että koko putkea pitää päivittää. (Wheeler 2023)

CI/CD putkia voidaan mahdollisesti käyttää, mutta kuin haastatteluissa todettiin, pitää ensin selvittää kehitystiimien tilanne automatisoinnin kannalta. On tärkeä

muistaa, että vaikka toimeksiannon tavoitteena on suunnitella työkalu, joka voi automaation avulla rakentaa testiympäristön, ei tämä työkalu voi pakottaa kehitystiimejä automatisoimaan heidän tekemistään. CI/CD putkia ei vaan voida ottaa käyttöön päättämällä niin, vaan pitää katsoa, että kehitystiimit kaikki käyttävät sitä. Multi-Repo lähetysmistapa voi olla todella hyvä ideana, mutta se pitää toimeksiantajan selvittää ensin, mikä on tämänhetkinen tilanne.

4.7 Pilviratkaisu

Testiympäristö pitää jonnekin asentaa ja mikä sen nykyaikaisempaa kuin pilvipalveluiden hyödyntäminen. Koska meiltä puuttuu tietoa Yritys X:n nykyistä pilviratkaisuista, voidaan me tämän opinnäytetyön myötä ehdottaa vaihtoehtoa, jota voisi hyödyntää tähän toimeksiantoon. Tähän toimeksiantoon voidaan hyödyntää esimerkiksi PaaS (Platform as a Service) ratkaisua. PaaS ratkaisussa, pilvipalvelun tarjoaja hoitaa taustalla olevan infrastruktuurin, servereiden toimintaa, tallennustilaa ja verkotusta. Yrityksen ei silloin tarvitse huomioida edellä mainittuja asioita, vaan voi enemmän keskittyä oman ohjelmansta kehittämiseen. (Lawal 2023, 37) Toisena mahdollisuutena on hyödyntää IaaS (Infrastructure as a Service) ratkaisua. Tämä ratkaisu antaa vielä enemmän vastuuta asiakkaalle, eli Yritys X:lle. IaaS ratkaisussa Yritys X olisi itse vastuussa infrastruktuurin konfiguroinnista. (Lawal, 31)

5 POHDINTA

Opinnäytetyön tavoitteena oli suunnitella testiympäristöjä rakentava automaatio, joka voisi rakentaa testiympäristön mahdollisimman yksinkertaisesti. Toimeksiannon tärkeimmäksi toiveeksi nousi työkalun yksinkertainen käyttö ja se, että sitä voisi käyttää henkilö, joka ei omaa suurta teknistä osaamista. Testiympäristöjen tärkeys nousi esille haastatteluissa, joka antoi ajatusta siitä, että miten hyvin tämän kaltainen automaatio kannattaa suunnitella. Opinnäytetyön tavoitteena oli luoda suunnitelma tästä työkalusta ja antaa toimeksiantajalle pohja minkä päälle toimeksiantaja voi lähteä rakentamaan tätä työkalua.

Toimeksiannon toteuttaminen oli yksi keskeisimmistä kysymyksistä opinnäytetyössä. Toimeksiantaja ei hakenut valmista teknistä toteutusta tähän opinnäytetyöhön, mutta suunnitella mahdollisimman selkeän kehityksen tälle automaatio työkalulle. Loin kaaviot, jotka mielestäni antoi sen mitä toimeksiannossa oli pyydetty. Mietin prosessi työkalun käyttöön ja annoin oman mielipiteeni arkkitehtuuriin. Löysin myös mahdollisia ratkaisuja, mitä tämän toimeksiannon toteutuksessa kannattaa miettiä esimerkiksi pilven ja CI/CD:n käytön näkökulmasta. Itselleni laaditut tutkimuskysymykset tämän toteuttamiseen on hieman jälkikäteen ajateltuna hankalia, koska en voi tarkalleen tietää kaikkea toimeksiantajasta, sekä nykytilanteesta.

Suurimmaksi haasteeksi toimeksiantajalle jää suuri määrä kehitystiimejä ja niiden nykytilanne. Toimeksiantajan pitäisi aloittaa projekti sillä, että kehitystiimeihin tehdään tiedustelut, että millä tavalla heillä käytetään automaattista komponenttien rakentamista tai vaikka CI/CD:n käyttöä. Automatisoitua työkalua on vaikea rakentaa, jos työkaluun tarvittavat komponentit eivät ole automatisoitu. Haastatteluissa nousi tämä aihe jo esille ja se oli selvästi jo jollain tasolla tietoista, että riittävästi informaatiota ei ole kehitystiimeistä ja heidän valmiudestansa. Tämä haaste oikeastaan vastaa jo kahteen kysymykseen, eli siihen mikä on se haaste ja mitä pitäisi seuraavaksi edistää toimeksiantajan päässä.

Yritys hyötyy tästä suunnitelmasta, koska siitä nähdään mitä tarvitaan tämän työkalun rakentamiseen ja nähdään nykyiset ongelmakohdat. Toimeksiantajan on tämän raportin myötä helpompi aloittaa projekti tämän suunnitelman pohjalta,

koska annan selkeän ja yksinkertaisen näkemyksen työkalun prosesseista ja rakenteesta.

Mielestäni vastaan alussa määriteltyihin tutkimuskysymyksiin niin hyvin kun oli näiden tietojen perusteella mahdollista. Testiympäristöjen automaattista rakentamisesta ei löydy hirveästi suoranaista dataa tai tietoa, josta voisi ottaa oppeja ja sama pätee myös toimeksiantajaa. Onnistuin kuitenkin tämän opinnäytetyön myötä suunnittelemaan pohjan testiympäristöjen rakentamisen automatisoinnin työkalulle.

LÄHTEET

- Aitman, B. 2016. Are you sure? How to write a confirmation dialog. Beth Altman. Viitattu 21.04.2024. <https://bethaitman.com/posts/ui-writing/confirmation/>.
- Fowler, M. & Scott, K. 1999. UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language. Boston: Addison Wesley.
- Hoffman, D. Grivel, E. & Battle L. 2005. Designing software architectures to facilitate accessible web applications. IBM Systems Journal, Volume 44 Issue 3.
- Hooda, I & Chhillar, R. 2015. Software Test Process, Testing Types and Techniques. International Journal of Computer Applications, Volume 111 No 13, 10–14.
- IBM. 2021. Use-case diagrams. Viitattu 21.4.2024. <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>
- International Software Testing Qualifications Board. Certified tester foundation level syllabus. Viitattu 22.4.2024 https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf
- Jyväskylän Yliopisto. Laadullinen tutkimus. 28.10.2021. Viitattu: 20.1.2024 <https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tutkimusstrategiat/laadullinen-tutkimus>.
- Krug, S. 2014. Don't make me think revisited. USA: New Riders
- LaunchDarkly. 2021. Test environment best practices: 8 you need to know. 16.4.2021. Viitattu 21.04.2024 <https://launchdarkly.com/blog/test-environment-best-practices-8-you-need-to-know/>
- Lawal, A. 2023. Cloud Computing Fundamentals. USA: Ojula Technology Innovations.
- Marget, A. 2024. Development and Test Environments: Understanding the Different Types of Environments. Unitrends. Viitattu 5.4.2024. <https://www.unitrends.com/blog/development-test-environments#:~:Text=without%20impacting%20users.-,Test%20Environment,Application%20or%20impair%20user%20experience>.
- Pathania, N. 2016. Learning Continuous Integration with Jenkins. Birmingham: Packt Publishing Ltd.
- Saaranen-Kauppinen, A. & Puusniekka, A. 2006. Mitä laadullinen tutkimus on: lyhyt oppimäärä. KvaliMOTV – Menetelmäopetuksen tietovaranto. Tampere: Yhteiskuntatieteellinen tietoarkisto. Viitattu: 5.4.2024 https://www.fsd.tuni.fi/menetelmaopetus/kvali/L1_2.html.
- Sacolick, I. 2018. What is CI/CD? Continuous integration and continuous delivery explained. San Mateo: Infoworld Media Group.

SurveyMonkey 2024. Laadullisen ja määrällisen tutkimuksen erot. Viitattu: 5.4.2024 <https://fi.surveymonkey.com/mp/quantitative-vs-qualitative-research/>.

The Investopedia Team. 2024. What is Cloud Computing. Investopedia. Viitattu 21.4.2024. <https://www.investopedia.com/terms/c/cloud-computing.asp>

Wheeler, A. 2023. Best Architecture for Dev Collaboration: Monorepo vs. Multirepo. GitKraken 23.8.2023. Viitattu: 28.4.2024 <https://www.gitkraken.com/blog/monorepo-vs-multi-repo-collaboration>.

Wysocki, R. 2019. Effective Project Management. Indianapolis: John Wiley & Sons, Inc.

LIITTEET

Liite 1. Haastattelukysymykset

Liite 1. Haastattelukysymykset

1. Mitä varten testiympäristöt ovat mielestäsi tärkeitä testauksessa?
2. Käytätkö itse testiympäristöjä aktiivisesti työssäsi ja näkyykö se sinun työssäsi kuinka tärkeänä?
3. Mitä hyötyjä tämän tyyppisestä automaatiosta olisi, joka rakentaa testiympäristön?
4. Jos tämän tyyppistä automaatiota ei tehtäisi, mitä luulet, että se voisi tarkoittaa tulevaisuudessa? Olisiko sillä suuria vaikutuksia?
5. Osaatko sanoa miten tämä teknisesti kannattaisi tehdä?
6. Voisiko tässä hyödyntää CI/CD putken käyttöä?
7. Onko mielessäsi jotain mitä pitäisi pitää mielessä, kun tätä automaatiota suunnitellaan?
8. Onko sinulla toiveita automaatiotyökalua kohtaan?
9. Millä tavalla itse työkalu pitäisi mielestäsi toteuttaa käyttöliittymän osalta? Tarvitaanko käyttöliittymää?
10. Kun automaatiota suunnitellaan, niin mikä on mielestäsi tärkein asia mihin pitäisi keskittyä?