OULUN AMMATTIKORKEAKOULU

Kalle Jämsä

VERSIOHALLINNAN INTEGROIMINEN JIRA-JÄRJESTELMÄÄN

VERSIOHALLINNAN INTEGROIMINEN JIRA-JÄRJESTELMÄÄN

Kalle Jämsä
Opinnäytetyö
Kevät 2022
Tradenomi (AMK), tietojenkäsittely
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tradenomi (AMK), tietojenkäsittely

Tekijä(t): Kalle Jämsä
Opinnäytetyön nimi: Versiohallinnan integroiminen Jira-järjestelmään
Työn ohjaaja(t): Pekka Ojala
Työn valmistumislukukausi ja -vuosi: helmikuu 2022 Sivumäärä: 26 + 1 liite

---

Opinnäytetyö toteutettiin toimeksiantona Detection Technology Oyj:lle (DT). Aiheena on parantaa ohjelmistotuotannon versiohallintaa integroimalla versiohallintatyökalu Jira-järjestelmään. Työskentelin kesän 2021 Detection Technology Oyj:llä Confluence Jira -pilottiprojektissa yhtenä järjestelmän ylläpitäjänä. Työsuhteen aikana huomattiin mahdollisuus versiohallinnan integroimisesta Jira-järjestelmään. Yrityksessä oli aikaisemmin jo havaittu tarve parantaa versio- ja lähdekoodihallintaa. Opinnäytetyössä käsiteltiin nykytila, mahdolliset työkalut, ominaisuudet sekä raportin tulokset.

Tavoitteina oli saada parempi kuva yrityksen versiohallinnasta, yhdenmukaistaa sitä ja ehdottaa ratkaisua integroitavaksi Jira-järjestelmään. Tutkimustyö tehtiin pääasiassa teemahaastatteluilla ja tutkimalla verkkolähteitä, joista osa on opinnäytetöitä. Pääaihe keskittyi tutkimaan eri työkalujen mahdollisuuksia ja testaamaan integraatiota.

Nykytila-analyysistä saa kattavan kuvan yrityksen tämänhetkisestä tilanteesta ja kehitystarpeista tulevaisuudelle. Se antoi hyvät lähtökohdat lähteä tutkimaan kahta työkalua integroitavaksi Jira-järjestelmään. Toimeksiantajan vaatimuksia ei päästy konkretisoimaan analyysivaiheessa. Työkaluista oli tarjolla monia englanninkielisiä verkkolähteitä, joten tietoa oli runsaasti saatavilla. Lopussa pääsimme integroimaan ja testaamaan molemmat työkalut.

Alustavasti valitussa ratkaisussa integroitiin GitLab-versionhallintatyökalu Jira-järjestelmään käyttäen API-rajapintaa sekä DVCS-liitintä. Yhdistämisen johdosta Jira-tehtäväkortilla pääsi näkemään versiohallintatietoja, kuten tallennukset, haarat ja yhdistämisen tilan. Kortilta pääsi navigoimaan linkin kautta kyseiseen tapahtumaan GitLab-järjestelmässä.

Kehitystyötä voisi jatkaa ottamalla käyttöön jatkuvan integraation ohjelmiston, kuten Jenkins, joka integroitaisiin Jira-järjestelmään. Lisäksi uskon, että ketterän kehityksen koulutuksista olisi tulevaisuudessa hyötyä Detection Technology Oyj:lle.

Asiasanat: Versiohallinta, Jira, lähdekoodi, Git, integraatio ja projektinhallinta.

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems

Author(s): Kalle Jämsä
Title of thesis: Version control integration to Jira
Supervisor(s): Pekka Ojala
Term and year when the thesis was submitted: Spring 2022
Number of pages: 26 + 1 appendix

This final thesis was created for Detection Technology Oyj (DT). Main topic was to get better understanding of company's version control and improve it by integrating tool to Jira system. I was working for Detection Technology in 2021 summer as Confluence Jira admin. During my employment, we got the idea for my thesis work. Thesis goes through current state, tool options and final decision proposal.

Target was to get a better understanding of the whole Detection Technology's version control, streamline it and propose a tool to be integrated into Jira system. Themed interviews were used to get the best understanding of the current state. Internet sources and testing were the biggest part of investigating tool options.

GitLab version control tool was selected and integrated during thesis for software and firmware development. Thesis reports detailed picture of DT's way of working and used tools. New positions were opened for the future evolution. DT got information of available tools and ideas to improve the version control process. Future improvements could contain CI (Continuous Integration) tool integration like Jenkins to be integrated into Jira system. Agile method trainings could be a powerful addition to existing training programs. Agile method training could contain parts of Scrum project management.

Keywords: version control, Jira, source code, Git, continuous integration, project management

SISÄLLYS

1 JOHDANTO

Opinnäytetyön taustalla oli Detection Technology Oyj:n (DT) tarve parantaa ja yhtenäistää ohjelmistokehitystä. DT osti pari vuotta sitten yrityksen, joka tuottaa ohjelmistojen ja algoritmien kehitystyötä. Versiohallinta ja sen työkalut olivat tulleet yritykseen työntekijöiden mukana, eikä niillä ole nimettyä omistajuutta. Työkalujen käytössä oli paljon vaihtelevuutta, ja niiden hallinta puuttui. Käynnissä oli projekti, jossa otettiin vaiheittain käyttöön Jira-projektinhallintatyökalu. Opintojen loppupuolella työskentelin kesän 2021 yrityksessä Jira-pääkäyttäjänä. Kesän aikana tarpeet ja työkalun kehitys korostuivat aika-ajoin. Työsuhteen aikana heräsi kiinnostus Jira-järjestelmän integraatiosta Git-työkaluun. Kesän jälkeen jatkoin työskentelyä yrityksessä ja aloin tutkia integraation mahdollisuuksia opinnäytetyönä. Opinnäytetyö on hyvä jatkumo työtehtävien ja ohjelmistokehityskoulutuksen lisäksi.

Opinnäytetyön ensimmäisessä vaiheessa pyritään saamaan kattava kuva DT:n versiohallinnan käytöstä ja sen työkaluista kaikissa toimipisteissä ympäri maailman. Toisessa vaiheessa tutkitaan työkalujen ominaisuuksia ja mahdollista integrointia Jira-järjestelmään. Viimeisessä vaiheessa raportoidaan tulokset ja dokumentoidaan ehdotettu ratkaisu.

Työssä käytettiin teemahaastatteluita nykytilan analysointiin sekä useita verkkolähteitä työkalujen tutkimiseen. Pääasiallisesti työkaluja koskevat tiedot on kerätty kehittäjän tai markkinoivan yrityksen verkkosivuilta. Verkkoaineisto painottui englanninkielisiin teksteihin. Aihetta on aikaisemmin käsitelty muun muassa Jänösen opinnäytetyössä ”Azure DevOps Git-versionhallintaominaisuuden integrointi Jira-ohjelmistoon” ja Tuurinkosken opinnäytetyössä ”Versionhallinnan työkalut ketterässä ohjelmistokehityksessä”.

Opinnäytetyön rakenne

Aikatauluksi työlle sovittiin kuusi kuukautta, jonka aikana toteutettiin kolme päävaihetta. Vaiheisiin kuului nykytila-analyysi, tutkimustyö sekä tulosten raportointi. Nykytila-analyysi toteutettiin pääasiassa haastattelemalla Jira- ja Git-käyttäjiä. Haastattelut toteutettiin teemahaastatteluina. Tutkimustyö keskittyi lähinnä kahden eri työkalun vertailuun internet-lähteiden ja testaamisen

kautta. Opinnäytetyön tulokset raportoidaan peilaten niitä ensimmäisessä vaiheessa määriteltyihin vaatimuksiin.

KUVIO 1.Opinnäytetyön rakenne

2 SANASTO

Git = Versiohallintajärjestelmä, jossa säilytetään ja hallinnoidaan ohjelmistokoodia.

Jira = Australialaisen Atlassian-yrityksen kehittämä projektinhallintaohjelmisto.

Firmware = (FW) on laiteohjelmisto, joka ohjaa laitteen toimintaa tai sen osaa.

Ohjelmisto = (Software, SW) on joukko käskyjä laitteen tai ohjelmiston käyttämiseen.

Algoritmi = Ohjelmistokoodia, joka pohjautuu vahvasti matematiikkaan ja logiikkaan.

DT = Detection Technology Oyj.

Issue = Jira-järjestelmän objekti, jolla voi olla useita eri tyyppejä, kuten tehtävä, ongelma tai tarina.

Jatkuva integraatio = **(**Continuous integration, CI) on ohjelmistotuotannon menetelmä, johon kuuluu erilaisia automaattisia testauksia.

Template = Ennalta määritelty sivu tai alusta, joka toimii aloituskohtana työlle.

Custom = Mukautettu tai mittatilauksena tehty työ tai suunnitelma.

Script = Tiedosto, jossa määritellään toimintoja, asetuksia tai määrityksiä.

Nightly build = CI-testit, jotka ajetaan päivän päätteeksi testaamaan päivän aikana tehdyt muutokset.

DevOps = Toimintamalli, joka pyrkii automatisoimaan ohjelmistokehitykseen, testaamiseen ja ylläpitoon liittyvät palvelut.

Docker = Tuoteperhe, jolla voidaan ajaa ohjelmistoa virtuaalisessa kontissa.

Repository = Ohjelmiston varastointipaikka palvelimella.

Haara = (Branch) on ohjelmistokehityksen sivu- tai päähaara, jossa voidaan kehittää ominaisuutta niin sanotusti sivussa.

Git LFS = (Large File Storage, LFS) on Atlassianin kehittämä lisäosa, joka helpottaa suurien tiedostojen, kuten kuvien tai videoiden, käsittelyä Git-varastossa (Atlassian Git LFS 2021c).

API = Ohjelmistorajapinta (Application Programming Interface, API).

Tallennus = (commit) Git-paketin tallennus palvelimelle.

Yhdistäminen = (Pull request / Merge request) Pyyntö haaran yhdistämisestä toiseen.

Dashboard = Ohjelmiston kojelauta, johon on kerätty dataa eri paikoista.

Workflow = Tehtävän työnkulku.

Hostaus = Palvelimen ylläpitäminen (host).

Runner = Virtuaalinen ohjelmaympäristö.

Iteraatio = Useaan kertaan tapahtuva toistaminen.

3 NYKYTILA-ANALYYSI

Nykytila-analyysillä haluttiin selvittää yrityksen versiohallinnan käyttöä eri käyttäjien ja toimipisteiden kesken. Yrityksessä tiedettiin, että käytössä on useita eri palvelimia sekä työkaluja. Teemahaastatteluiden avulla käytiin aihetta läpi seitsemän haastateltavan kanssa. Lisäksi tulevaisuutta ja integraation tarpeita käytiin läpi käyttäjien kanssa.

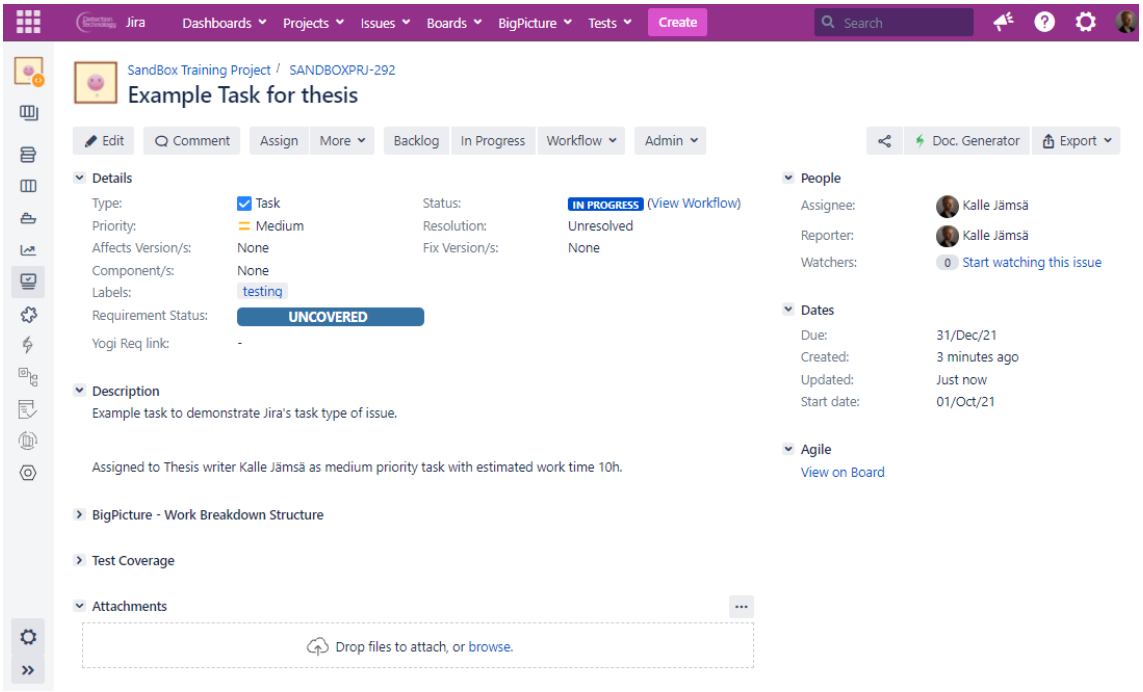
Detection Technology (DT) on julkinen osakeyhtiö, joka on perustettu vuonna 1991. DT kehittää, valmistaa, markkinoi ja myy röntgenkuvantamislaitteiden komponentteja ja järjestelmiä lääketieteen, turvallisuusalan ja teollisuuden laitteisiin. Yhtiöllä on toimipisteet Suomessa, Kiinassa, Ranskassa ja Yhdysvalloissa. Tuotanto on pääasiallisesti keskitetty Kiinan tehtaisiin. Tuotekehitystä toteutetaan Suomessa, Kiinassa ja Ranskassa. DT listautui 2015 First North -pörssiin nimellä DETEC. Työntekijöitä yhtiössä on hieman yli 400, joista valtaosa työskentelee Kiinassa. (Kinnunen 2021.)

3.1 Järjestelmät

Git on hajautettu versionhallintajärjestelmä, jossa varastoidaan lähdekoodia. Git on kehitetty Helsingissä 1990-luvulla alun perin Linux-kehitykseen. On kehitetty lukuisia työkaluja, jotka helpottavat käyttöä lisäämällä käyttöliittymän Git-versionhallinnalle. Tunnetuimmat työkalut ovat esimerkiksi GitHub, BitBucket, Tortoise Git, SmartGit ja GitLab. Ilman työkalua Git-versionhallintaa voi käyttää vain komentorivin kautta. Varasto Git-järjestelmässä on repositorio (Repository) ja yksittäinen koodin lisääminen tai tallennus on commit.

Jira on ohjelmistoja tuottavan australialaisen Atlassian-yrityksen tuottama projektinhallintajärjestelmä, joka on pääasiallisesti kehitetty ohjelmistotuotannon kehitykseen ja projektinhallintaan. Jira-järjestelmän ensimmäinen versio julkaistiin vuonna 2005. DT:llä on käytössä Jira-palvelinversio. Järjestelmä otetaan yrityksessä täysin käyttöön opinnäytetyön kirjoittamisen aikana. Keskeisin toiminnallisuus Jira-järjestelmässä on tehtävienhallinta (Issue Management) erilaisissa tauluissa ja projekteissa. Issue on yhteinen nimitys usealle erilaiselle tehtävätyypille, kuten epicille, tehtävälle, tarinalle, bugille tai testille. Tehtävällä voi olla useita eri

kenttiä, jotka varastoivat tietoa kyseisestä tehtävästä. Kuviossa 1 on esimerkki Task-tyypin tehtävästä Sandbox-projektissa.



KUVIO 2. Kuvakaappaus Jira-tehtäväkortista

3.2 Haastattelut

Haastatteluissa käytiin läpi haastateltava ja tämän pääasiallinen nimike tai työtehtävät. Kysymyksiä oli Git-työkalun käytöstä, ominaisuuksista, vahvuuksista ja heikkouksista. Tulevaisuuden kehitysideoita pohdittiin myös suurimmassa osassa haastatteluista. Haastateltavilta kysyttiin edellisten työsuhteiden aikana hyväksi todetuista työkaluista ja käytänteistä. Usein haastateltavat mielellään kertoivat avoimesti aiheesta, jopa ilman kysymyksiä. Haastattelupohja löytyy liitteestä 1.

Haastatteluita tehtiin yhteensä seitsemän. Osa haastatteluista toteutettiin etäyhteyden kautta haastateltavan toimipisteen takia. Viisi haastattelua tehtiin englannin kielellä. Kaikki haastattelut nauhoitettiin, ja niiden tallenteet ovat vapaasti DT:n käytössä. Haastattelut toteutettiin teemahaastatteluina. Teema- tai puolistrukturoitu haastattelu on keskustelunomainen tilanne, jossa käydään läpi ennalta suunniteltuja teemoja. Haastateltavien kanssa ei välttämättä puhuta asioista samassa laajuudessa. Lisäksi teemojen järjestys voi muuttua. Haastattelijalla on yleensä hallussaan tiiviit muistiinpanot käsiteltävistä teemoista. Muistiinpanoja voi tehdä aiheista,

kysymyksistä tai avainsanoista. Teemahaastattelun ei siis tulisi olla tarkkojen kysymysten esittämistä tarkassa järjestyksessä. Teemoista ja niiden alateemoista pyritään keskustelemaan varsin vapaasti. (Saaranen-Kauppinen & Puusniekka 2006.)

3.3 Vaatimukset ja nykytila

Alusta lähtien suurin tunnistettu vaatimus työkalulle oli palvelinversio eli niin sanottu on-premise, joka mahdollistaa työkalun käytön ja hallinnan DT:n omalla serverillä sisäverkossa. Palvelimen asennus oli tärkeää Kiinan toimistojen kannalta, koska tiedettiin, että pilvipalvelut olivat tuottaneet haasteita aikaisemmin. Asennus tehtäisiin jo olemassa olevalle palvelimelle. Lisäksi työkalu on Git-pohjainen ja siinä on mahdollisuuksia Jira-integraatiolle. Jatkuvan integraation (CI) mahdollisuudet ovat tulevaisuuden kannalta tärkeitä, ja jos niitä käytetään työkalussa, niiden pitää olla tarpeeksi kustomoitavissa. CI-ominaisuuksiin pitää pystyä määrittämään erikseen ohjelmisto, firmware ja laitteisto.

Haastattelussa todettiin seuraavasti:
”Pelkkä template-käyttö ei riitä.”
”Niin kauan kuin tehdään custom-softaa, pitää pystyä tekemään custom-testejä."

Subversion

Subversion on toinen versionhallintajärjestelmä Git-järjestelmän rinnalla. Subversion eli SVN-järjestelmä on ollut DT:llä käytössä jo ennen Git-pohjaista versiohallintaa. Opinnäytetyössä ei keskitytä tämän työkalun käyttöön tai kehitykseen, mutta on tärkeää tietää, mikä se on. DT:llä SVN on käytössä edelleen tuotannossa, ja siellä säilytetään tiettyjä asetustiedostoja (Script). SVN-järjestelmästä siirryttiin osittain Git-järjestelmään, koska Git-pohjaiset järjestelmät ovat kevyempiä: lähdekoodia on helpompi käsitellä ja nopeampi yhdistää. Haastattelussa kerrotaan: ”Osa asetustiedostoista tulee pysymään SVN-järjestelmässä, koska ne on rakennettu niin syvälle sisään DT:n ohjelmistoihin. ” Aikaisemmin kaikki ohjelmistot olivat SVN-järjestelmässä.

Git-työkalut

Jokaisella DT:n toimipisteellä on käytössä ainoastaan tai osittain Git-työkalu nimeltä Gitea. Gitea on avoimen lähdekoodin Git-pohjainen versiohallintaohjelmisto. Haastatteluissa käyttäjät eivät olleet tyytyväisiä Gitea-työkalun käyttöön ja ominaisuuksiin. Haastatteluissa ilmeni, että Gitea ei ole riittävä työkalu ominaisuuksien puutteellisuuksien takia.

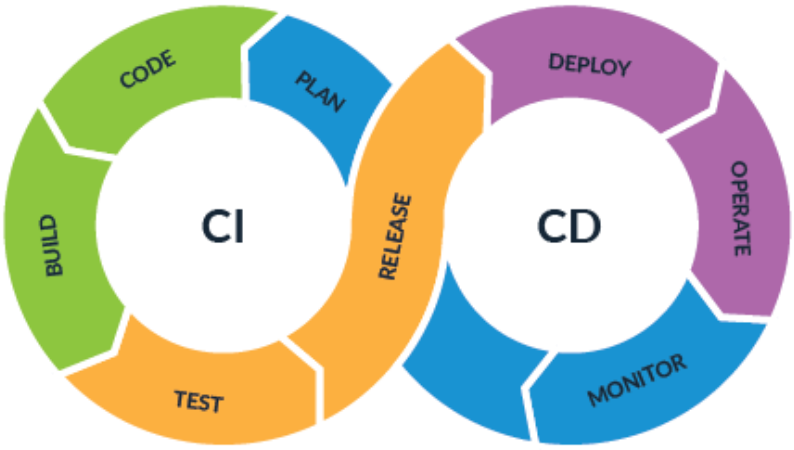
Gitea-työkalua kommentoitiin muun muassa seuraavasti:
"Giteassa ei ole niitä ominaisuuksia, joita tulevaisuudessa tarvitaan."
”Gitea ei ole tarpeeksi hyvä.”
”Gitea ei ole käyttäjäystävällinen.”

Gitea-työkalun puutteellisuuden kommentoitiin olevan pienen yrityksen ongelma. Yhdessä haastattelussa kerrottiin, että Gitea-ohjelmiston suojauksessa on ongelmia, jonka takia ohjelmistosta pitäisi luopua. Gitea on asennettu DT:n palvelimelle yhden ohjelmistokehittäjän puolesta hänen tullessa DT:lle töihin.

GitLab otettiin käyttöön noin vuosi sitten, ja se on toinen DT:llä käytettävistä versiohallintatyökaluista. Sitä käytetään Gitea-ohjelmistoa vähemmän, vaikka GitLab-ohjelmistoa kehutaan hyväksi ja riittäväksi. GitLab on käytössä ohjelmistokehityksessä ainakin Suomessa. Palvelin pyörii irrallisena muista palvelimista tavallisessa tornitietokoneessa. GitLab-ohjelmiston ongelmana on omistajuuden puutos, joka johtaa siihen, että ohjelmistokehittäjien pitää käyttää aikaa palvelimen ylläpitoon. Kehittäjien ajanpuutteen vuoksi palvelin on riittämättömällä ylläpidolla, ja kärsii esimerkiksi kovalevyongelmista. Palvelimella käytetään ohjelmia (runner) testaamaan ohjelmistoa fyysisellä laitteistolla. GitLab-konfiguraatiota kommentoidaan haastatteluissa seuraavasti: ”GitLab ajaa dockkerin sisällä GitLab runnereita, jotka ajavat dockereita dockerin sisällä eli docker in docker.” ”Virtualisoi virtualisaation sisällä, mikä aiheuttaa omia ongelmia.". GitLab-ohjelmiston toiminnoista kerrotaan seuraavasti: ”Käytetään normaalitoimintoja kuten pull requestia, jolla tehty release, build, kloonattu, testattu ja CI-scriptauksia.”. GitLab-järjestelmän ominaisuuksiin kuuluu CI-testit, joita käytetään vaihtelevalla menestyksellä osittain ajanpuutteen vuoksi. Testien ylläpito vaatii runsaasti aikaa ohjelmistokehittäjiltä.

Jatkuva integraatio

Jatkuva integraatio (Continuous integration, CI) on prosessi, jolla automatisoidaan useiden osallistujien koodimuutosten integrointi yhdeksi ohjelmistoksi. CI perustuu versiohallintaan ja automaattisiin toimintoihin. Yksinkertaistettuna CI on useita toimintoja, joissa testataan ohjelmistopalasien yhteensopivuutta. Usein ohjelmistoprojekteissa kehitetään useita eri ominaisuuksia yhtä aikaa, joten niiden yhteensopivuuksien testaaminen on tärkeä osa projektia. Jatkuva integraatio (CI) kulkee yhdessä jatkuvan julkaisun kanssa (Continuous Deployment, CD). Menetelmät mainitaan yleensä yhdessä, koska ne ovat jatkumoa toisilleen prosessissa. CD painottuu enemmän testien jälkeiseen julkaisuun ja jakamiseen. CI/CD-työkaluja ovat esim. Jenkins ja Bamboo. Osa Git-työkaluista pitää sisällään CI/CD-ominaisuuksia. Pääperiaate näillä käytänteillä ja työkaluilla on lisätä tehokkuutta pienemmällä vaivalla eli tehdä enemmän vähemmällä (Garay 2021).



KUVIO 3. CI/CD-virtaus (Garay 2021)

Hallinta ja roolit

Detection Technology Oyj:n pääasiallinen toimiala on elektroniikka ja komponentit, minkä johdosta ohjelmistonhallinta ja –suunnittelu on olleet hieman taka-alalla. Ohjelmistokehitys tapahtuu hieman eri tyylillä yrityksessä, jossa painopisteenä on em. asiat. Ohjelmistokehittäjä kuvaa DT:n ohjelmistokehitystyyliä vanhaksi: moderneja työkaluja ei ole käytettävissä. Haastatteluissa kerrotaan, että: ”Se on täysipäiväinen työ ottaa haltuun SW ja FW kehittäjien päivittäiset testit (nightly build) ja niiden ylläpito".

Lisäksi haastatteluista ilmenee seuraavaa:
DT:ltä puuttuu ohjelmistokehityksestä rooleja, kuten ohjelmistoarkkitehti ja DevOps-insinööri.
Ohjelmistotiimi kaipaisi kehityksen avuksi parempaa tehtävienhallintaa ja käytetyille ohjelmistoille omistajuutta.
Järjestelmien omistajuudelle ja ylläpidolle on suuri tarve, sillä ne vaativat aikaa ja Linux-pohjaisien järjestelmien osaamista.

4 TYÖKALUT

Opinnäytetyön tutkimukseen päätettiin valita kaksi suosittua Git-työkalua, jotta tutkimus ja testaaminen eivät jakautuisi liian moneen eri vaihtoehtoon. GitLab valittiin, koska se on usealle DT:n työntekijälle ennestään tuttu ja hyväksi todettu työkalu. BitBucket valittiin, sillä se on saman yrityksen tuotantoa kuin integroitava järjestelmä Jira, joten sen integrointimahdollisuudet ovat laajat. Gitea-ohjelmisto päätettiin jättää tutkimuksista ulos haastatteluissa tulleiden tietojen pohjalta.

Työkalu yhdistetään Jira-järjestelmään suoraan asetuksista tai mahdollisesti käyttämällä kauppapaikalta ladattavaa lisäosaa, joka mahdollistaa integraation. Esimerkiksi Jänösen opinnäytetyössä on käytetty lisäosaa nimeltä Git integration for Jira", jolla voi nähdä koodimuutokset Jira-järjestelmässä. Lisäosa luo linkin koodimuutoksen ja tietyn Jira-tehtävän välille. Lisäosa lisää tehtävänäkymään Git Roll Up- ja Git Commits -välilehdet. Lisäosalla voidaan hallita haaroja ja tehdä vetopyyntöjä. Edellä mainittu lisäosa on maksullinen, ja sen hinta riippuu käyttäjämäärästä. (Jänönen 2021.)

Jatkuvan integraation työkalu Jenkins voidaan integroida Jira-järjestelmään käyttämällä kauppapaikalta ladattavaa lisäosaa. Lisäosan avulla Jira-tehtävällä voidaan nähdä esimerkiksi testien tuloksia, varoituksia epäonnistuvista testeistä ja Jenkins-tietoja kootusti ohjelmiston julkaisun yhteydessä. Lisäosa mahdollistaa automaattisten sääntöjen luomisen ennalta annettujen sääntöjen ja tapahtumien mukaan. Lisäosia em. käyttöön on tutkittu useita, ja suosituin maksaa yhden dollarin per käyttäjä vuodessa. Tutkittu lisäosa on nimeltään "Jenkins Integration for Jira". Päätoiminnallisuus on visualisoida Jenkins-ohjelmiston tapahtumat poistumatta Jira-järjestelmästä. (Atlassian Marketplace - Jenkins Integration for Jira 2021e.)

4.1 GitLab

GitLab on avoimen lähdekoodin monipuolinen versiohallintatyökalu. GitLab Inc. julkaisi työkalun ensimmäisen version vuonna 2014. GitLab-ohjelmistoa pidetään open-core-ohjelmistona, koska sen käyttö on vapaata ja ilmaista, mutta maksamalla voi avata lisää ominaisuuksia. StackShare sivuston mukaan GitLab on maailman suosituin ”on-premise”-Git-työkalu. (Stackshare GitLab 2021b.)

GitLab-ohjelmiston parhaat puolet olivat käyttäjien äänestyksen mukaan:

- itse hallittava palvelin
- kattava ilmaisversio
- yhteisöversio ilman tukea
- helppo käyttöliittymä
- CI-työkalut ja niiden integraatiomahdollisuudet. (Stackshare GitLab 2021b.)

Työkalun ominaisuudet

Keskeisimmät GitLab-ohjelmiston ominaisuudet verrattuna BitBucket-työkaluun ovat CI/CD-työkalut, tehtävienhallinta ja avoimen lähdekoodin tuomat edut. (GitLab Feature Comparison 2021b.) GitLab-ohjelmisto pitää sisällään yhteensä 55 ominaisuutta. (GetApp GitLab 2021b.)

GitLab-ilmaisversion keskeisimmät ominaisuudet listattuna:
- CI/CD-kanavat (pipeline) ja lähdekoodin palat (Snippets)
- varasto- ja projektinhallinta
- haara (branch) oikeudet ja liitoksien tarkistukset (merge)
- Jira-integraatio ja kolmannen osapuolen työkaluihin
- API:t ja Git LFS 2.0 tuki.

GitLab-ohjelmiston maksullisten versioiden tuomat ominaisuudet listattuna:

- iteraatiot
- lisätoiminnallisuudet tehtävienhallintaan
- kooditarkastelun lisäominaisuudet, kuten säännöt
- laajempi käyttö- ja pääsyoikeuksien hallinta
- roolit, kuten koodin omistaja ja varmistettu tallentaja
- tiedostopohjat ja koodin etsiminen
- ympärivuorokautinen tuki
- LDAP-ryhmien synkronointi
- Jira-tehtävien listaus.

Jira-integraation mahdollisuudet

Jira GitLab -integraatio on mahdollista toteuttaa suoraan liittämällä järjestelmät käyttäen Jira-järjestelmän DVCS-yhdistämistä (distributed version control system). DVCS on versiohallinnalle tarkoitettu liitin, jossa syötetään Jira-järjestelmälle tietoja, kuten yhdistettävä palvelin, nimi ja autentikointitiedot. DVCS-liitintä pidetään vanhana teknologiana, ja osa Git-ohjelmistoista on kehittänyt korvaavan applikaation Jira-pilviversion integraatioon. Esimerkiksi GitLab ja GitHub omaavat Jira-pilviversiolle omat applikaationsa. Jira-järjestelmässä automaattiset tehtävämuutokset eivät ole mahdollisia ilman lisäosaa.

Jira GitLab -integraatio suositellaan toteutettavaksi erillisen lisäosan kanssa, koska se lisää kontrollia ja helpottaa integraation hallintaa. Lisäosat ovat maksullisia, ja hinnat liikkuvat kymmenistä euroista pariin tuhanteen euroon. Atlassian kauppapaikka tarjoaa useita eri vaihtoehtoja ladattavaksi Jira-ympäristöön. (GitLab Jira integration 2021c.)

Suosituin lisäosa nimeltä ”Git Integration for Jira” tarjoaa seuraavia ominaisuuksia:

- varastojen (repository) katselmointi Jira-järjestelmässä
- versioiden ja tallennuksien (commit) vertailu
- analyyttiset taulukot ja kaaviot
- Jira-tehtävien koonti Git-tallennusten mukaan
- varastojen massamuutokset
- tallennuksien etsiminen JQL-kielellä.

GitLab-integraatio Jenkins-ohjelmistoon on mahdollista toteuttaa lisäosan avulla. Lisäosan ovat kehittäneet molempien ohjelmistojen käyttäjät, joten se ei ole virallisesti kummankaan tukema. Kehittäjät suosivat Jenkins-ohjelmistoa sen avoimen lähdekoodin takia. Se mahdollistaa kattavan muokattavuuden ja laajennusmahdollisuuden yli 300 ohjelmistoon. (Educba Comparison 2021.)

Lisäosan perustoiminnot:

- käynnistää Jenkins-testejä koodien tallennuksen tai yhdistämisien mukaan
- näyttää testien tuloksia GitLab-ohjelmistossa.

Lisäosan edistyneet toiminnot:

- aloittaa tai jättää pois testejä riippuen haaran nimestä
- käyttää nimikkeitä (Tag) käynnistämään testit
- hyväksyä yhdistäminen (merge) automaattisesti testien läpimenon seurauksena.

Hinnasto

GitLab-hinnasto määräytyy valitusta tilauksen tasosta. Ilmaisversiossa ei ole rajattu käyttäjien määrää. Lisäminuutteja CI-testaukseen ja tallennustilaa voi myös ostaa erikseen. Maksulliset tilaukset avaavat toimintoja, kuten sääntöjen hallintaa, analytiikkaa, kattavaa tehtävienhallintaa, rooleja ja lähdekoodin etsintää. (GitLab Pricing 2021a.)

GitLab-ohjelmiston tilaustasot:

1. Ilmainen (Free)
2. Korotettu (Premium)
3. Laajin (Ultimate)

TAULUKKO 1. GitLab-ohjelmiston hinnasto (GitLab Pricing 2021a.)

Käyttäjämäärä	*Premium USD $ /v*	*Premium EUR € /v*	*Ultimate USD $ /v*	*Ultimate EUR € /v*
1	19	17	99	86
25	475	415	2 475	2 159
50	950	829	4 950	4 319
100	1 900	1 658	9 900	8 638

USD:n ja EUR:n vaihtokurssi on haettu 11.11.2021 ja EUR-hinnat on pyöristetty kokonaisluvuiksi.

Testaus

Jira-järjestelmä integroitiin lopulta DT:n palvelimelle asennettuun GitLab-ohjelmistoon käyttäen DVCS-liitintä sekä API-integraatiota. Molemmat järjestelmät toimivat sisäverkossa sekä DT:n omalla palvelimella. GitLab-integraation testaamiseen käytettiin lopulta paljon enemmän aikaa ja resursseja alustavan työkaluvalinnan takia. Maksullisen lisäosan käyttöä harkittiin, mutta päätettiin toteuttaa integraatio ilman sitä. GitLab-ohjelmiston luotettujen sertifikaattien tallentaminen tuotti ongelmia alussa järjestelmän bugin vuoksi. API-integraatio mahdollisti Jira-tehtävien listauksen GitLab-järjestelmän sisälle. DVCS-liittimen kautta tallennuksen (push) yhteydessä voi kommentoida, kirjata aikaa ja vaihtaa tehtävän tilaa. API-integraatiota varten Jira-järjestelmään luotiin käyttäjä, jonka kautta GitLab sai oikeudet Jira-järjestelmään.

4.2 BitBucket

Atlassian osti BitBucket-ohjelmiston vuonna 2010, ja vuonna 2011 palvelu alkoi tukea Git-ohjelmistoa. Entinen nimitys ohjelmistosta on Stash. Palvelusta on tarjolla uusiin tilauksiin pilvi- sekä datakeskusversiot. Uusien palvelinversioiden myynti lopetettiin helmikuussa 2021, ja niiden tuki loppuu helmikuussa 2024. Monipuolista integraatiota mainostetaan Atlassian-tuoteperheen vahvuutena. Pilviversiota on mahdollista käyttää pienellä tiimillä ilmaiseksi, mutta käyttö muuttuu maksulliseksi, kun käyttäjämäärä kasvaa yli viiden. Ilmainen käyttö mahdollistaa testauksen tässä opinnäytetyössä ilman työkalun ostamista. Vuonna 2019 Atlassian uutisoi BitBucket-blogissaan yli 10 miljoonan käyttäjän ja yli 28 miljoonan varaston (repository) rajapyykin ylityksestä. (Atlassian BitBucket Features 2021b.)

BitBucket-ohjelmiston parhaat puolet käyttäjien äänestyksessä:

- ilmaiset salatut varastot (repository)
- yksinkertainen asennus
- miellyttävä käyttöliittymä
- edullinen Git-hostaus
- laajat API- ja palveluintegraatiot. (Stackshare BitBucket 2021)

Työkalun ominaisuudet

BitBucket-ohjelmiston vahvuuksiin kuuluu yksinkertainen käyttöliittymä, johon on lisätty hyödyllisiä ominaisuuksia, kuten historian hallinta ja erilaiset analytiikkaominaisuudet. Osa ominaisuuksista, kuten CI-automatisointi, on kaupallistettu erilliseksi Atlassian-työkaluksi. Yksityisten varastojen (repository) määrää ei ole rajoitettu työkalussa. BitBucket tukee Atlassian kauppapaikan lisäosia, joten toimintoja pystyy lisäämään tarpeen vaatiessa. (Atlassian BitBucket Features 2021b.)

Atlassian BitBucket-datakeskuksen ominaisuudet listattuna:
- varaston- ja projektinhallinta
- branch oikeudet ja liitoksien tarkistukset (merge)
- integraatiot Jira-järjestelmään ja kolmannen osapuolen ohjelmistoihin
- koodin etsiminen, analytiikka ja ryhmittely (clustering)
- API:t ja Git LFS.

Pilvi- ja datakeskusversiot jakavat suurimman osan ominaisuuksista ja toiminnoista. Suurin eroavaisuus näiden välillä syntyy tiedon, lisenssien ja erilaisten säädöksien hallinasta, koska datakeskusversiossa näiden hallinta jää asiakkaalle. Pilviversion laskutuksessa on enemmän joustavuutta, ja se mahdollistaa esimerkiksi kuukausilaskutuksen. CI-työkalut eivät ole mukana datakeskusversiossa, ja Atlassian suosittelee CI-työkaluaan nimeltä Bamboo. Atlassian tuoteperheen CI-työkalu Bamboo on saatavilla datakeskusversiona. Se on mahdollista yhdistää BitBucket- ja Jira-ohjelmistoon ilman lisäosia. Bamboo-ohjelmiston pääasialliset toiminnot ovat CI- ja CD-automaatio. Bamboo on maksullinen työkalu, joka toteuttaa testejä työnkulun määräämässä järjestyksessä. Työkalu sisältää myös useita raportti- ja analytiikkaominaisuuksia, jotka tarkkailevat testien tuloksia ja koodin laatua. (Atlassian Bamboo 2021d.)

Integraatiomahdollisuudet

Työkalujen integraatiossa on mahdollista linkittää Jira-tehtävä ja Git-tallennus (commit), mikä mahdollistaa tiedon jakamisen näiden välillä. Linkin jälkeen Jira-tehtäväkortilla voi nähdä branch-, commit- ja pull request -tiedot. Atlassian mainostaa sivullansa, että tiimit, jotka integroivat Jira- ja BitBucket-järjestelmät, julkaisevat koodia 14 % useammin. Työkalut on mahdollista integroida toisiinsa ilman mitään ulkopuolista lisäosaa. Integrointi tehdään järjestelmän asetuksista lisäämällä yhdistävä linkki. Integraatio Jira-järjestelmään on vahva ja yksinkertainen käyttää, koska järjestelmät jakavat monia samoja elementtejä sekä termejä, kuten workflow ja dashboard. GetApp-sivuston mukaan BitBucket on integroitavissa 145:een eri järjestelmään, joista suurimpia ovat Slack, Trello, Zoom ja Jira. (GetApp BitBucket Integrations 2021a.) Aiemmin työssä esiteltyä lisäosaa ”Git Integration for Jira” voi lisäksi käyttää BitBucket-ohjelmiston kanssa. Integraation johdosta voi luoda automaattisia tapahtumia, kuten vaihtaa tehtävän tilatietoa, kun uusi haara luodaan.

Jira-palvelinversion ja BitBucket-datakeskuksen integraatio-ominaisuudet:

- ajantasainen kehitystieto tehtäväkortilla
- tehtävän tilamuutos BitBucket-tapahtumalla
- version kehityksen monitorointi Jira-järjestelmässä
- Jira-tehtävän käyttäminen BitBucket-palvelimella
- sääntöjen asettaminen (esimerkiksi tallennuksen täytyy sisältää X).

BitBucket-ohjelmisto on mahdollista integroida myös Jenkins-työkaluun. Integraatio suositellaan toteutettavaksi käyttäen kauppapaikalta ladattavaa lisäosaa. Lisäosia on useita vaihtoehtoja. Ohjeissa Jenkins-ohjelmiston kehittäjät suosittelevat lisäosaa nimeltä ”Jenkins integration for Bitbucket Server”, jonka on kehittänyt Atlassian. (Jenkins Bitbucket Server Integration 2021.)

Lisäosa mahdollistaa seuraavat ominaisuudet:

- automaattisten käynnistysten luonti
- varastojen automaattinen kloonaus testejä varten
- datan hakeminen ohjelmistojen välillä
- testitulosten tuonti BitBucket-järjestelmään.

Hinnasto

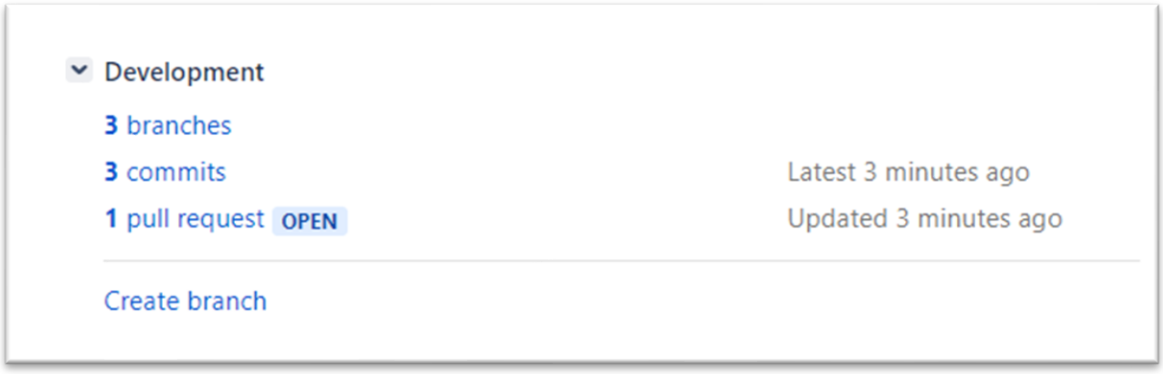
Data Center eli itsehallintana olevan BitBucket-ohjelmiston hinnoittelu riippuu käyttäjämäärästä. Hinnoittelu esitellään alla olevassa taulukossa. Data Center-vaihtoehto tarjotaan vuosittaisella laskutuksella, johon kuuluu päivitykset ja tuki. Atlassian tarjoaa datakeskustuotteillensa kokeilujakson. Vertailuna BitBucket-pilviversion saa 50 käyttäjälle 1 550 eurolla vuodessa. (Atlassian BitBucket 2021a.)*TAULUKKO 2. BitBucket*-ohjelmiston *hinnasto* (Atlassian BitBucket 2021a.)

Käyttäjämäärä	*USD $ vuodessa*	*EUR € vuodessa*
25	2 300	1 980
50	4 200	3 617
100	7 600	6 546

USD:n ja EUR:n vaihtokurssi on haettu 11.11.2021 ja EUR-hinnat on pyöristetty kokonaisluvuiksi.

Testaus

Integraation testaaminen toteutettiin BitBucket-ohjelmiston pilviversiossa resurssien puutteen vuoksi. Osa ominaisuuksista voi vaihdella palvelin- ja pilviversion välillä. Järjestelmien yhdistämiseen käytettiin DVCS-liitintä syöttämällä käyttäjä- ja autentikointitiedot Jira-järjestelmälle. Liittimen lisääminen onnistui nopeasti, ja liitos järjestelmien välillä oli toiminnassa välittömästi. Jira-tehtäväkortilta pystyy luomaan uuden kehityshaaran BitBucket-järjestelmään. Näin toimittaessa Jira-tehtävä on suoraan linkitetty oikeaan haaraan, joten mahdollisuutta kirjoitusvirheelle ei ole. Tallennuksen (commit) yhteydessä on mahdollisuus kommentoida, kirjata työtunteja tai päivittää tehtävän tilatieto. Jira-tehtävän päivitykset pitää syöttää älykäs tallennus -syntaksilla (Smart Commit Syntax). DVCS-liitin synkronoi tiedot järjestelmien välillä 60 minuutin välein, joten kehitystiedot eivät ole jatkuvasti ajan tasalla. Alla olevassa kuvankaappauksessa on versiohallintatiedot linkkeineen: linkkiä painamalla saa avattua ikkunan, jossa näkyy tarkemmat tiedot kyseisestä tapahtumasta.
Esimerkki päivitystiedosta: “KEY-123 #comment started working on the issue #in-progress #time 10h”. (GitLab 2021d.)



KUVIO 4. Kuvakaappaus versiohallintatiedoista Jira-kortilla

5 TULOKSET

Työkalun valintaa ja integraatiota kiirehdittiin suuren tarpeen vuoksi, joten valintaa ei tehty järjestelmien testaamisen tuloksilla. GitLab-järjestelmä valittiin DT:n käyttöön, koska jatkuva integraatio oli toteutettavissa järjestelmän sisällä ja koska työkalu oli ennestään tuttu monille työntekijöille. GitLab asennettiin Suomeen ensisijaisena palvelimena, ja toisaalle asennettiin kaksi tukevaa palvelinta helpottamaan tiedostojen siirtoaikoja esimerkiksi Kiinan etäisyyden vuoksi.

GitLab-ohjelmiston integraatio Jira-järjestelmään mahdollistaa paremman tehtävienhallinnan ohjelmistokehityksessä lisäämällä linkin tehtävän ja lähdekoodin välille. Integraatio toteutettiin käyttämällä integraatioon API- sekä DVCS-linkkiä (distributed version control system). Laajemmat integraatio-ominaisuudet on kuvattu opinnäytetyön luvussa GitLab 3.1. Uusi GitLab-järjestelmä sai nimetyn hallinnon ja tiimin ylläpitämään järjestelmää.

Opinnäytetyön aikana DT avasi paikat ohjelmistoarkkitehdille ja CI/QA johtajalle. DT:n panostus nousee jatkuvasti kehittyvien markkinoiden takia opinnäytetyössä käsiteltäviin asioihin. GitLab-ohjelmistoa esiteltiin ja koulutettiin henkilöstölle opinnäytetyön lopussa.

6 POHDINTA

Opinnäytetyö eteni viikoittaisen suunnitelman mukaisesti, ja jokaiseen työvaiheeseen päästiin määräajassa. Ohjaajan kanssa pidettiin palaverit joko ennakkoon suunnitellusti tai tarvittaessa. Palavereissa käytiin läpi kysymykset, ongelmat ja työn eteneminen. Alun perin tarkoitus oli työskennellä opinnäytetyön parissa 50 % työajasta, mutta joinain viikkoina tämä oli haastavaa toteuttaa muiden töiden ohella. En kokenut, että tämä olisi merkittävästi vaikuttanut työhön. Kaikkiin päämääriin päästiin aikataulussa, joten muutoksia suunniteltuun aikatauluun ei tehty. Keskimääräisesti toteutunut työmäärä oli noin 40 % työajasta. Suunniteltu aikataulu ja viikkokohtainen aiheiden suunnittelu helpottivat opinnäytetyön toteutuksessa.

Osa tutkittavista aiheista tai testeistä riippui edellisen vaiheen etenemisestä. Niinpä yhden vaiheen vaihtuessa toiseen pidettiin ohjaajan ja toimeksiantajan kanssa palaveri, jossa suunniteltiin seuraavan vaiheen etenemistä. Esimerkiksi työn alussa toimeksiantaja ei päässyt määrittelemään tarkasti vaatimuksia, joten tutkimusvaiheessa vaatimuksien tarkkuus hieman vaihtui. Tähän pystyttiin vastaamaan nopeasti, joten työn eteneminen pystyttiin turvaamaan normaalisti. Hyvän etenemisen vuoksi opinnäytetyöhön lisättiin loppuvaiheessa Jenkins-ohjelmiston tarkastelua.

Yhteistyö ohjaajan ja toimeksiantajan edustajien kanssa sujui erinomaisesti, ja koin, että jokainen osapuoli oli hyvin tietoinen työn etenemisestä. Työ perustettiin ns. projektina, jossa opinnäytetyön kirjoittajana toimin projektipäällikön roolissa. Aikataulua, muistiinpanoja, palaverin kirjanpitoja ja muita dokumentteja hallittiin Confluence-järjestelmässä, jossa ne olivat kaikkien saatavilla.

Jatkaisin kehitystä järjestelmien integraatiolla Jenkins-ohjelmistoon, jota uusi CI/CD-johtaja hallitsee. Tulevaisuudessa Jira-järjestelmässä voisi nähdä versiohallintatiedot sekä jatkuvan integraation tulokset. Ohjelmistotuotannon kehitys vaatii jatkuvaa työtä, jos halutaan käyttää tehokaita työtapoja ja -kaluja. Ketterän työtavan implementointi vaatii tiimeille sekä tiimien vetäjille koulutuksia.

Pääsin opinnäytetyön aikana oppimaan ja tuntemaan työntekijöitä sekä toimeksiantajan yritystä. Ihmiset olivat innostuneita kertomaan ajatuksiaan ja pitivät siitä, että heitä kuunnellaan. Tutustumisen ohella pääsin ratkaisemaan ongelmia yhteistyössä, ja uskon, että tämä tulee

auttamaan jatkossakin työtehtävissäni. Motivaatio ja asenne auttoivat paljon työn edistymisessä ja kuljettivat prosessin läpi.

LÄHTEET

Atlassian BitBucket 2021a. BitBucket Overview. Hakupäivä 2.11.2021. https://www.atlassian.com/enterprise/data-center/bitbucket.

Atlassian BitBucket Features 2021b. BitBucket Features. Hakupäivä 24.11.2021. https://bitbucket.org/product/features.

Atlassian Git LFS 2021c. Git LFS. Hakupäivä 2.11.2021. https://www.atlassian.com/git/tutorials/git-lfs.

Atlassian Bamboo 2021d. Bamboo Overview. Hakupäivä 11.4.2021. https://confluence.atlassian.com/bamboo.

Atlassian Marketplace - Jenkins Integration for Jira 2021e. Jenkins Integration for Jira. Hakupäivä 20.12.2021. https://marketplace.atlassian.com/apps/1211376/jenkins-integration-for-jira?hosting=server&tab=overview.

Educba Comparison 2021. Gitlab CI vs Jenkins. Hakupäivä 27.12.2021. https://www.educba.com/gitlab-ci-vs-jenkins/.

Garay, Daniel 2021. Parasoft - Implementing QA in a CI/CD Pipeline. Hakupäivä 27.10.2021. https://www.parasoft.com/blog/implementing-qa-in-a-ci-cd-pipeline/.

GetApp BitBucket Integrations 2021a. BitBucket Integrations. Hakupäivä 9.11.2021. https://www.getapp.com/it-management-software/a/bitbucket/integrations/.

GetApp GitLab 2021b. GitLab product overview. Hakupäivä 29.11.2021. https://www.getapp.com/it-management-software/a/gitlab/.

GitLab Pricing 2021a. GitLab Pricing. Hakupäivä 29.11.2021. https://about.gitlab.com/pricing/.

GitLab Feature Comparison 2021b. GitLab Self-Managed Feature Comparison. Hakupäivä 29.11.2021. https://about.gitlab.com/pricing/self-managed/feature-comparison/.

GitLab Jira integration 2021c. Jira integrations. Hakupäivä 29.11.2021. https://docs.gitlab.com/ee/integration/jira/.

GitLab Jira DVCS 2021d. Jira DVCS connector. Hakupäivä 30.11.2021. https://docs.gitlab.com/ee/integration/jira/dvcs.html.

Jenkins Bitbucket Server Integration 2021. Bitbucket Server Integration. Hakupäivä 13.12.2021. https://plugins.jenkins.io/atlassian-bitbucket-server-integration/.

Jänönen, Matti 2021. Azure DevOps Git-versionhallintaominaisuuden integrointi Jira-ohjelmistoon. Opinnäytetyö. Kajaani: KAMK. https://urn.fi/URN:NBN:fi:amk-2021052812231.

Kinnunen, Juha 2021. Inderes.fi - Detection Technology. Hakupäivä 19.10.2021. https://www.inderes.fi/fi/yhtiot/detection-technology.

Saaranen-Kauppinen, Anita & Puusniekka, Anna 2006. KvaliMOTV - Menetelmäopetuksen tietovaranto – Teemahaastattelu. Hakupäivä 13.10.2021. https://www.fsd.tuni.fi/menetelmaopetus/kvali/L6_3_2.html.

Stackshare BitBucket 2021. Bitbucket Overview. Hakupäivä 2.11.2021. https://stackshare.io/bitbucket.

Stackshare GitLab 2021b. GitLab Overview. Hakupäivä 30.11.2021. https://stackshare.io/gitlab.

HAASTATTELUKYSYMYKSET LIITE 1

Aihe: Projektinhallinta ja Git-linkkaus / Versiohallinta Git-käyttö

Haastattelija: Kalle Jämsä

Haastateltava:

Pvm:

Haastattelumuoto: Teemahaastattelu

Kysymykset:

Voidaanko nauhoittaa myöhempää käyttöä varten?

Kuka olet, ja mikä on pääasiallinen työtehtäväsi tai nimikkeesi?

Hallitsetko projekteja, joissa käytetään versionhallintaa ja nyt tai tulevaisuudessa Jiraa?

Kerro työstäsi Jirassa tai Git-työkalussa.

Käytätkö työssäsi versionhallintatyökalua?

Kerro työkalun käytöstäsi?

Questions:

Who are you what you do?

Your past with Git?

What are you using Gitlab Gitea etc.?

Have you used Bitbucket?

Have you used Jira?

What features are important in Git tool?

What would you like to see with Jira integration?

Projektin hallinta / Jira

Käyttö:

Kuinka usein? Päivittäin/kerran viikossa/kerran kuukaudessa?

Hallitsetko haaroja tai muita versioita?

Osallistutko useisiin projekteihin, joissa on käytössä Git?

Jiran tärkeimmät ominaisuudet versionhallinnan osalta?

Kommentointi?

Issue link to version?

Versioidenhallinta Git-tehtävään?

Bugin linkkaaminen versioon?

Version status?

CI-info?

Versiotiedot issuessa (pulls, commits yms?)

Versionhallinta / Git

JENKINS (käyttö, kokemukset)

Käyttö:

Kuinka usein? Päivittäin/kerran viikossa/kerran kuukaudessa?

Hallitsetko branchejä tai muita versioita?

Osallistutko useisiin projekteihin, joissa on käytössä Git?

Työkalu:

Mitä työkalua käytät?

UI vai komentorivi?

Onko työkalulla suuri merkitys?

Eniten käytetyt ominaisuudet?

Parhaat ja huonoimmat asiat tällä hetkellä käytössä olevassa työkalussa?

Puuttuuko siitä jotain?